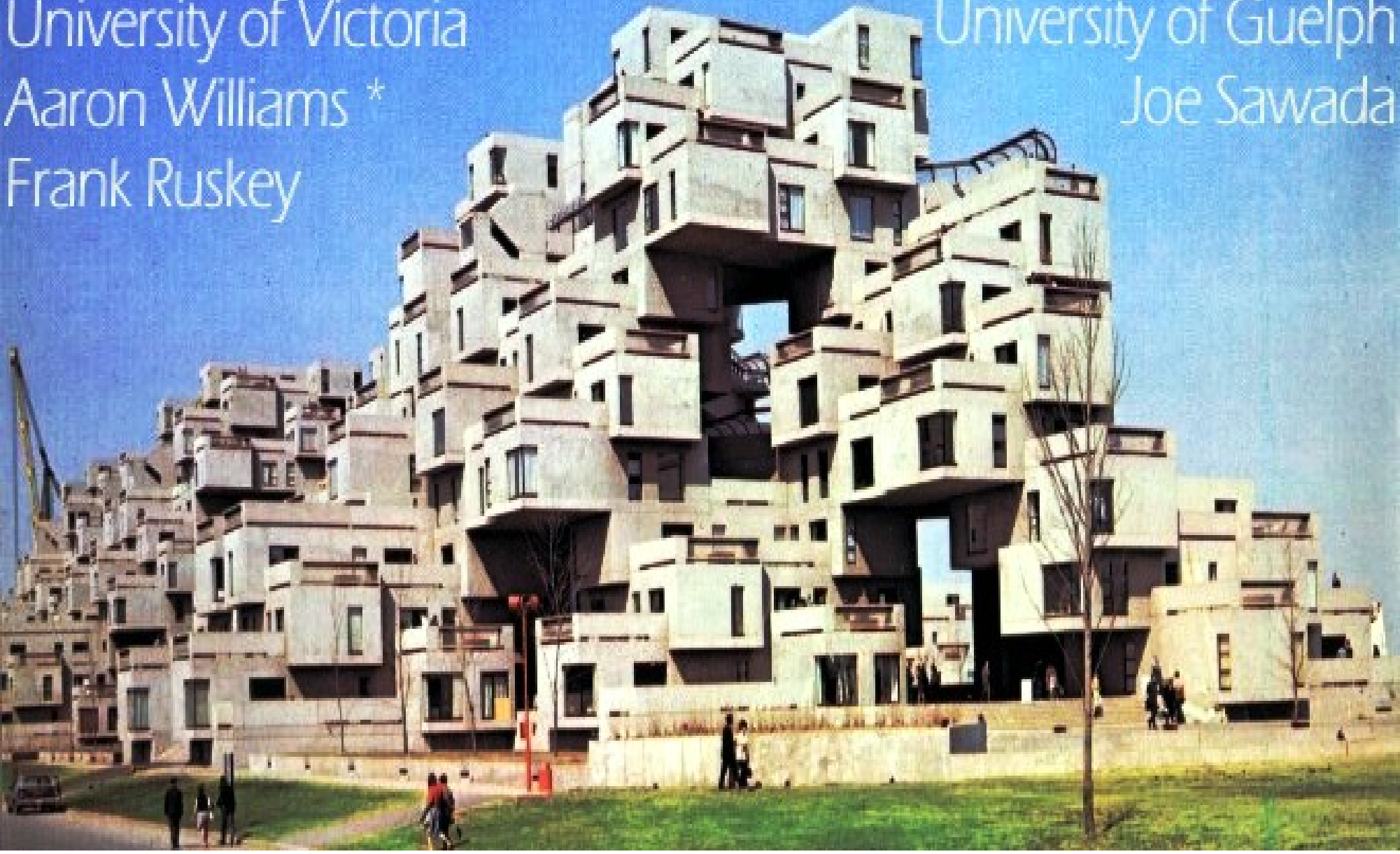
Fixed-Density de Bruijn Cycles

University of Victoria
Aaron Williams *
Frank Ruskey

University of Guelph
Joe Sawada



de Bruijn Cycles

Let $\mathbb{B}(N)$ be the set of length N binary strings.

$$|\mathbb{B}(N)| = 2^N.$$

A *de Bruijn cycle* is a binary string of length 2^N that contains every string in $\mathbb{B}(N)$ exactly once as a circular substring. For example,

$$0000100110101111$$

is a de Bruijn cycle for $N = 4$ since it contains the following substrings

$$0000, 0001, 0010, 0100, 1001, 0011, 0110, 1101,$$
$$1010, 0101, 1011, 0111, 1111, 1110, 1100, 1000$$

that include every string in $\mathbb{B}(4)$ exactly once.

Theorem (de Bruijn 1946)

de Bruijn cycles exist for all N.

de Bruijn cycles can be efficiently constructed using the lexicographic order of prime-prefixes of necklaces.

de Bruijn Cycles

Let $\mathbb{B}(N)$ be the set of length N binary strings.

$$|\mathbb{B}(N)| = 2^N.$$

A *de Bruijn cycle* is a binary string of length 2^N that contains every string in $\mathbb{B}(N)$ exactly once as a circular substring. For example,

$$0000100110101111$$

is a de Bruijn cycle for $N = 4$ since it contains the following substrings

$$0000, 0001, 0010, 0100, 1001, 0011, 0110, 1101,$$
$$1010, 0101, 1011, 0111, 1111, 1110, 1100, 1000$$

that include every string in $\mathbb{B}(4)$ exactly once.

Theorem (de Bruijn 1946)

de Bruijn cycles exist for all N.

de Bruijn cycles can be efficiently constructed using the lexicographic order of prime-prefixes of necklaces.

de Bruijn Cycles

Let $\mathbb{B}(N)$ be the set of length N binary strings.

$$|\mathbb{B}(N)| = 2^N.$$

A *de Bruijn cycle* is a binary string of length 2^N that contains every string in $\mathbb{B}(N)$ exactly once as a circular substring. For example,

$$0000100110101111$$

is a de Bruijn cycle for $N = 4$ since it contains the following substrings

$$0000, 0001, 0010, 0100, 1001, 0011, 0110, 1101,$$
$$1010, 0101, 1011, 0111, 1111, 1110, 1100, 1000$$

that include every string in $\mathbb{B}(4)$ exactly once.

Theorem (de Bruijn 1946)

de Bruijn cycles exist for all N.

de Bruijn cycles can be efficiently constructed using the lexicographic order of prime-prefixes of necklaces.

de Bruijn Cycles

Let $\mathbb{B}(N)$ be the set of length N binary strings.

$$|\mathbb{B}(N)| = 2^N.$$

A *de Bruijn cycle* is a binary string of length 2^N that contains every string in $\mathbb{B}(N)$ exactly once as a circular substring. For example,

$$0000100110101111$$

is a de Bruijn cycle for $N = 4$ since it contains the following substrings

$$0000, 0001, 0010, 0100, 1001, 0011, 0110, 1101,$$
$$1010, 0101, 1011, 0111, 1111, 1110, 1100, 1000$$

that include every string in $\mathbb{B}(4)$ exactly once.

Theorem (de Bruijn 1946)

de Bruijn cycles exist for all N.

de Bruijn cycles can be efficiently constructed using the lexicographic order of prime-prefixes of necklaces.

de Bruijn Cycles

Let $\mathbb{B}(N)$ be the set of length N binary strings.

$$|\mathbb{B}(N)| = 2^N.$$

A *de Bruijn cycle* is a binary string of length 2^N that contains every string in $\mathbb{B}(N)$ exactly once as a circular substring. For example,

$$0000100110101111$$

is a de Bruijn cycle for $N = 4$ since it contains the following substrings

$$0000, 0001, 0010, 0100, 1001, 0011, 0110, 1101,$$
$$1010, 0101, 1011, 0111, 1111, 1110, 1100, 1000$$

that include every string in $\mathbb{B}(4)$ exactly once.

Theorem (de Bruijn 1946)

de Bruijn cycles exist for all N.

de Bruijn cycles can be efficiently constructed using the lexicographic order of prime-prefixes of necklaces.

de Bruijn Cycles

Let $\mathbb{B}(N)$ be the set of length N binary strings.

$$|\mathbb{B}(N)| = 2^N.$$

A *de Bruijn cycle* is a binary string of length 2^N that contains every string in $\mathbb{B}(N)$ exactly once as a circular substring. For example,

$$0000100110101111$$

is a de Bruijn cycle for $N = 4$ since it contains the following substrings

$$0000, 0001, 0010, 0100, 1001, 0011, 0110, 1101,$$
$$1010, 0101, 1011, 0111, 1111, 1110, 1100, 1000$$

that include every string in $\mathbb{B}(4)$ exactly once.

Theorem (de Bruijn 1946)

de Bruijn cycles exist for all N.

de Bruijn cycles can be efficiently constructed using the lexicographic order of prime-prefixes of necklaces.

de Bruijn Cycles

Let $\mathbb{B}(N)$ be the set of length N binary strings.

$$|\mathbb{B}(N)| = 2^N.$$

A *de Bruijn cycle* is a binary string of length 2^N that contains every string in $\mathbb{B}(N)$ exactly once as a circular substring. For example,

$$0000100110101111$$

is a de Bruijn cycle for $N = 4$ since it contains the following substrings

$$0000, 0001, 0010, 0100, 1001, 0011, 0110, 1101,$$
$$1010, 0101, 1011, 0111, 1111, 1110, 1100, 1000$$

that include every string in $\mathbb{B}(4)$ exactly once.

Theorem (de Bruijn 1946)

de Bruijn cycles exist for all N.

de Bruijn cycles can be efficiently constructed using the lexicographic order of prime-prefixes of necklaces.

de Bruijn Cycles

Let $\mathbb{B}(N)$ be the set of length N binary strings.

$$|\mathbb{B}(N)| = 2^N.$$

A *de Bruijn cycle* is a binary string of length 2^N that contains every string in $\mathbb{B}(N)$ exactly once as a circular substring. For example,

$$0000\mathbf{10}0110101111$$

is a de Bruijn cycle for $N = 4$ since it contains the following substrings

$$0000, 0001, \mathbf{0010}, 0100, 1001, 0011, 0110, 1101,$$
$$1010, 0101, 1011, 0111, 1111, 1110, 1100, 1000$$

that include every string in $\mathbb{B}(4)$ exactly once.

Theorem (de Bruijn 1946)

de Bruijn cycles exist for all N.

de Bruijn cycles can be efficiently constructed using the lexicographic order of prime-prefixes of necklaces.

de Bruijn Cycles

Let $\mathbb{B}(N)$ be the set of length N binary strings.

$$|\mathbb{B}(N)| = 2^N.$$

A *de Bruijn cycle* is a binary string of length 2^N that contains every string in $\mathbb{B}(N)$ exactly once as a circular substring. For example,

$$0000\mathbf{0100}1101011111$$

is a de Bruijn cycle for $N = 4$ since it contains the following substrings

$$0000, 0001, 0010, \mathbf{0100}, 1001, 0011, 0110, 1101,$$
$$1010, 0101, 1011, 0111, 1111, 1110, 1100, 1000$$

that include every string in $\mathbb{B}(4)$ exactly once.

Theorem (de Bruijn 1946)

de Bruijn cycles exist for all N.

de Bruijn cycles can be efficiently constructed using the lexicographic order of prime-prefixes of necklaces.

de Bruijn Cycles

Let $\mathbb{B}(N)$ be the set of length N binary strings.

$$|\mathbb{B}(N)| = 2^N.$$

A *de Bruijn cycle* is a binary string of length 2^N that contains every string in $\mathbb{B}(N)$ exactly once as a circular substring. For example,

$$0000\mathbf{1001}10101111$$

is a de Bruijn cycle for $N = 4$ since it contains the following substrings

$$0000, 0001, 0010, 0100, \mathbf{1001}, 0011, 0110, 1101,$$
$$1010, 0101, 1011, 0111, 1111, 1110, 1100, 1000$$

that include every string in $\mathbb{B}(4)$ exactly once.

Theorem (de Bruijn 1946)

de Bruijn cycles exist for all N.

de Bruijn cycles can be efficiently constructed using the lexicographic order of prime-prefixes of necklaces.

de Bruijn Cycles

Let $\mathbb{B}(N)$ be the set of length N binary strings.

$$|\mathbb{B}(N)| = 2^N.$$

A *de Bruijn cycle* is a binary string of length 2^N that contains every string in $\mathbb{B}(N)$ exactly once as a circular substring. For example,

$$0000100110101111$$

is a de Bruijn cycle for $N = 4$ since it contains the following substrings

$$0000, 0001, 0010, 0100, 1001, 0011, 0110, 1101,$$
$$1010, 0101, 1011, 0111, 1111, 1110, 1100, 1000$$

that include every string in $\mathbb{B}(4)$ exactly once.

Theorem (de Bruijn 1946)

de Bruijn cycles exist for all N.

de Bruijn cycles can be efficiently constructed using the lexicographic order of prime-prefixes of necklaces.

de Bruijn Cycles

Let $\mathbb{B}(N)$ be the set of length N binary strings.

$$|\mathbb{B}(N)| = 2^N.$$

A *de Bruijn cycle* is a binary string of length 2^N that contains every string in $\mathbb{B}(N)$ exactly once as a circular substring. For example,

$$0000\color{red}{0110}1011\color{gray}{11}$$

is a de Bruijn cycle for $N = 4$ since it contains the following substrings

$$0000, 0001, 0010, 0100, 1001, 0011, \color{red}{0110}, \color{gray}{1101,}$$
$$\color{gray}{1010, 0101, 1011, 0111, 1111, 1110, 1100, 1000}$$

that include every string in $\mathbb{B}(4)$ exactly once.

Theorem (de Bruijn 1946)

de Bruijn cycles exist for all N.

de Bruijn cycles can be efficiently constructed using the lexicographic order of prime-prefixes of necklaces.

de Bruijn Cycles

Let $\mathbb{B}(N)$ be the set of length N binary strings.

$$|\mathbb{B}(N)| = 2^N.$$

A *de Bruijn cycle* is a binary string of length 2^N that contains every string in $\mathbb{B}(N)$ exactly once as a circular substring. For example,

$$0000100\color{red}{1101}01111$$

is a de Bruijn cycle for $N = 4$ since it contains the following substrings

$$0000, 0001, 0010, 0100, 1001, 0011, 0110, \color{red}{1101},$$
$$1010, 0101, 1011, 0111, 1111, 1110, 1100, 1000$$

that include every string in $\mathbb{B}(4)$ exactly once.

Theorem (de Bruijn 1946)

de Bruijn cycles exist for all N.

de Bruijn cycles can be efficiently constructed using the lexicographic order of prime-prefixes of necklaces.

de Bruijn Cycles

Let $\mathbb{B}(N)$ be the set of length N binary strings.

$$|\mathbb{B}(N)| = 2^N.$$

A *de Bruijn cycle* is a binary string of length 2^N that contains every string in $\mathbb{B}(N)$ exactly once as a circular substring. For example,

$$0000100110101111$$

is a de Bruijn cycle for $N = 4$ since it contains the following substrings

$$0000, 0001, 0010, 0100, 1001, 0011, 0110, 1101,$$
$$1010, 0101, 1011, 0111, 1111, 1110, 1100, 1000$$

that include every string in $\mathbb{B}(4)$ exactly once.

Theorem (de Bruijn 1946)

de Bruijn cycles exist for all N.

de Bruijn cycles can be efficiently constructed using the lexicographic order of prime-prefixes of necklaces.

de Bruijn Cycles

Let $\mathbb{B}(N)$ be the set of length N binary strings.

$$|\mathbb{B}(N)| = 2^N.$$

A *de Bruijn cycle* is a binary string of length 2^N that contains every string in $\mathbb{B}(N)$ exactly once as a circular substring. For example,

$$0000100110101111$$

is a de Bruijn cycle for $N = 4$ since it contains the following substrings

$$0000, 0001, 0010, 0100, 1001, 0011, 0110, 1101,$$
$$1010, 0101, 1011, 0111, 1111, 1110, 1100, 1000$$

that include every string in $\mathbb{B}(4)$ exactly once.

Theorem (de Bruijn 1946)

de Bruijn cycles exist for all N.

de Bruijn cycles can be efficiently constructed using the lexicographic order of prime-prefixes of necklaces.

de Bruijn Cycles

Let $\mathbb{B}(N)$ be the set of length N binary strings.

$$|\mathbb{B}(N)| = 2^N.$$

A *de Bruijn cycle* is a binary string of length 2^N that contains every string in $\mathbb{B}(N)$ exactly once as a circular substring. For example,

$$0000100110101111$$

is a de Bruijn cycle for $N = 4$ since it contains the following substrings

$$0000, 0001, 0010, 0100, 1001, 0011, 0110, 1101,$$
$$1010, 0101, 1011, 0111, 1111, 1110, 1100, 1000$$

that include every string in $\mathbb{B}(4)$ exactly once.

Theorem (de Bruijn 1946)

de Bruijn cycles exist for all N.

de Bruijn cycles can be efficiently constructed using the lexicographic order of prime-prefixes of necklaces.

de Bruijn Cycles

Let $\mathbb{B}(N)$ be the set of length N binary strings.

$$|\mathbb{B}(N)| = 2^N.$$

A *de Bruijn cycle* is a binary string of length 2^N that contains every string in $\mathbb{B}(N)$ exactly once as a circular substring. For example,

$$0000100110101111$$

is a de Bruijn cycle for $N = 4$ since it contains the following substrings

$$0000, 0001, 0010, 0100, 1001, 0011, 0110, 1101,$$
$$1010, 0101, 1011, 0111, 1111, 1110, 1100, 1000$$

that include every string in $\mathbb{B}(4)$ exactly once.

Theorem (de Bruijn 1946)

de Bruijn cycles exist for all N.

de Bruijn cycles can be efficiently constructed using the lexicographic order of prime-prefixes of necklaces.

de Bruijn Cycles

Let $\mathbb{B}(N)$ be the set of length N binary strings.

$$|\mathbb{B}(N)| = 2^N.$$

A *de Bruijn cycle* is a binary string of length 2^N that contains every string in $\mathbb{B}(N)$ exactly once as a circular substring. For example,

$$0000100110101111$$

is a de Bruijn cycle for $N = 4$ since it contains the following substrings

$$0000, 0001, 0010, 0100, 1001, 0011, 0110, 1101,$$
$$1010, 0101, 1011, 0111, 1111, 1110, 1100, 1000$$

that include every string in $\mathbb{B}(4)$ exactly once.

Theorem (de Bruijn 1946)

de Bruijn cycles exist for all N.

de Bruijn cycles can be efficiently constructed using the lexicographic order of prime-prefixes of necklaces.

de Bruijn Cycles

Let $\mathbb{B}(N)$ be the set of length N binary strings.

$$|\mathbb{B}(N)| = 2^N.$$

A *de Bruijn cycle* is a binary string of length 2^N that contains every string in $\mathbb{B}(N)$ exactly once as a circular substring. For example,

$$0000100110101111$$

is a de Bruijn cycle for $N = 4$ since it contains the following substrings

$$0000, 0001, 0010, 0100, 1001, 0011, 0110, 1101,$$
$$1010, 0101, 1011, 0111, 1111, 1110, 1100, 1000$$

that include every string in $\mathbb{B}(4)$ exactly once.

Theorem (de Bruijn 1946)

de Bruijn cycles exist for all N.

de Bruijn cycles can be efficiently constructed using the lexicographic order of prime-prefixes of necklaces.

de Bruijn Cycles

Let $\mathbb{B}(N)$ be the set of length N binary strings.

$$|\mathbb{B}(N)| = 2^N.$$

A *de Bruijn cycle* is a binary string of length 2^N that contains every string in $\mathbb{B}(N)$ exactly once as a circular substring. For example,

$$0000100110101111$$

is a de Bruijn cycle for $N = 4$ since it contains the following substrings

$$0000, 0001, 0010, 0100, 1001, 0011, 0110, 1101,$$
$$1010, 0101, 1011, 0111, 1111, 1110, 1100, 1000$$

that include every string in $\mathbb{B}(4)$ exactly once.

Theorem (de Bruijn 1946)

de Bruijn cycles exist for all N.

de Bruijn cycles can be efficiently constructed using the lexicographic order of prime-prefixes of necklaces.

de Bruijn Cycles

Let $\mathbb{B}(N)$ be the set of length N binary strings.

$$|\mathbb{B}(N)| = 2^N.$$

A *de Bruijn cycle* is a binary string of length 2^N that contains every string in $\mathbb{B}(N)$ exactly once as a circular substring. For example,

$$0000100110101111$$

is a de Bruijn cycle for $N = 4$ since it contains the following substrings

$$0000, 0001, 0010, 0100, 1001, 0011, 0110, 1101,$$
$$1010, 0101, 1011, 0111, 1111, 1110, 1100, 1000$$

that include every string in $\mathbb{B}(4)$ exactly once.

Theorem (de Bruijn 1946)

de Bruijn cycles exist for all N.

de Bruijn cycles can be efficiently constructed using the lexicographic order of prime-prefixes of necklaces.

de Bruijn Cycles

Let $\mathbb{B}(N)$ be the set of length N binary strings.

$$|\mathbb{B}(N)| = 2^N.$$

A *de Bruijn cycle* is a binary string of length 2^N that contains every string in $\mathbb{B}(N)$ exactly once as a circular substring. For example,

$$0000100110101111$$

is a de Bruijn cycle for $N = 4$ since it contains the following substrings

$$0000, 0001, 0010, 0100, 1001, 0011, 0110, 1101,$$
$$1010, 0101, 1011, 0111, 1111, 1110, 1100, 1000$$

that include every string in $\mathbb{B}(4)$ exactly once.

Theorem (de Bruijn 1946)

de Bruijn cycles exist for all N.

de Bruijn cycles can be efficiently constructed using the lexicographic order of prime-prefixes of necklaces.

de Bruijn Cycles

Let $\mathbb{B}(N)$ be the set of length N binary strings.

$$|\mathbb{B}(N)| = 2^N.$$

A *de Bruijn cycle* is a binary string of length 2^N that contains every string in $\mathbb{B}(N)$ exactly once as a circular substring. For example,

$$0000100110101111$$

is a de Bruijn cycle for $N = 4$ since it contains the following substrings

$$0000, 0001, 0010, 0100, 1001, 0011, 0110, 1101,$$
$$1010, 0101, 1011, 0111, 1111, 1110, 1100, 1000$$

that include every string in $\mathbb{B}(4)$ exactly once.

Theorem (de Bruijn 1946)

de Bruijn cycles exist for all N.

de Bruijn cycles can be efficiently constructed using the lexicographic order of prime-prefixes of necklaces.

de Bruijn Cycles

Let $\mathbb{B}(N)$ be the set of length N binary strings.

$$|\mathbb{B}(N)| = 2^N.$$

A *de Bruijn cycle* is a binary string of length 2^N that contains every string in $\mathbb{B}(N)$ exactly once as a circular substring. For example,

$$0000100110101111$$

is a de Bruijn cycle for $N = 4$ since it contains the following substrings

$$0000, 0001, 0010, 0100, 1001, 0011, 0110, 1101,$$
$$1010, 0101, 1011, 0111, 1111, 1110, 1100, 1000$$

that include every string in $\mathbb{B}(4)$ exactly once.

Theorem (de Bruijn 1946)

de Bruijn cycles exist for all N.

de Bruijn cycles can be efficiently constructed using the lexicographic order of prime-prefixes of necklaces.

de Bruijn Cycles

Let $\mathbb{B}(N)$ be the set of length N binary strings.

$$|\mathbb{B}(N)| = 2^N.$$

A *de Bruijn cycle* is a binary string of length 2^N that contains every string in $\mathbb{B}(N)$ exactly once as a circular substring. For example,

$$0000100110101111$$

is a de Bruijn cycle for $N = 4$ since it contains the following substrings

$$0000, 0001, 0010, 0100, 1001, 0011, 0110, 1101,$$
$$1010, 0101, 1011, 0111, 1111, 1110, 1100, 1000$$

that include every string in $\mathbb{B}(4)$ exactly once.

Theorem (de Bruijn 1946)

de Bruijn cycles exist for all N.

de Bruijn cycles can be efficiently constructed using the lexicographic order of prime-prefixes of necklaces.

de Bruijn Cycles

Let $\mathbb{B}(N)$ be the set of length N binary strings.

$$|\mathbb{B}(N)| = 2^N.$$

A *de Bruijn cycle* is a binary string of length 2^N that contains every string in $\mathbb{B}(N)$ exactly once as a circular substring. For example,

$$0000100110101111$$

is a de Bruijn cycle for $N = 4$ since it contains the following substrings

$$0000, 0001, 0010, 0100, 1001, 0011, 0110, 1101,$$
$$1010, 0101, 1011, 0111, 1111, 1110, 1100, 1000$$

that include every string in $\mathbb{B}(4)$ exactly once.

Theorem (de Bruijn 1946)

de Bruijn cycles exist for all N.

de Bruijn cycles can be efficiently constructed using the lexicographic order of prime-prefixes of necklaces.

de Bruijn Cycles

Let $\mathbb{B}(N)$ be the set of length N binary strings.

$$|\mathbb{B}(N)| = 2^N.$$

A *de Bruijn cycle* is a binary string of length 2^N that contains every string in $\mathbb{B}(N)$ exactly once as a circular substring. For example,

$$0000100110101111$$

is a de Bruijn cycle for $N = 4$ since it contains the following substrings

$$0000, 0001, 0010, 0100, 1001, 0011, 0110, 1101,$$
$$1010, 0101, 1011, 0111, 1111, 1110, 1100, 1000$$

that include every string in $\mathbb{B}(4)$ exactly once.

Theorem (de Bruijn 1946)

de Bruijn cycles exist for all N.

de Bruijn cycles can be efficiently constructed using the lexicographic order of prime-prefixes of necklaces.

Concatenation and Lexicographic Order

If $\mathbf{s} \in \mathbb{B}(N)$ then s_i refers to the ith bit of $\mathbf{s}$ for $1 \leq i \leq N$.

The *concatenation* of two strings $\mathbf{s} \in \mathbb{B}(N)$ and $\mathbf{t} \in \mathbb{B}(M)$ is

$$\mathbf{s} \cdot \mathbf{t} = s_1 s_2 \cdots s_N t_1 t_2 \cdots t_M = \mathbf{st}.$$

Exponentiation represents repeated concatenation of the same string.

$$\mathbf{s}^k = \underbrace{\mathbf{s} \cdot \mathbf{s} \cdots \mathbf{s}}_{\text{k copies}}$$

Binary strings of equal length are totally ordered by their numerical value. That is, if $\mathbf{s}, \mathbf{t} \in \mathbb{B}(N)$ then $\mathbf{s} < \mathbf{t}$ if there exists i such that

$$s_1 s_2 \cdots s_i = t_1 t_2 \cdot t_i \text{ and } s_{i+1} = 0 \text{ and } t_{i+1} = 1.$$

$\mathcal{L}(\mathbb{B})$ denotes this *lexicographic order* for any $\mathbb{B} \subseteq \mathbb{B}(N)$. For example,

$$\mathcal{L}(\mathbb{B}(3)) = 000, 001, 010, 011, 100, 101, 110, 111.$$

Concatenation and Lexicographic Order

If $\mathbf{s} \in \mathbb{B}(N)$ then s_i refers to the ith bit of $\mathbf{s}$ for $1 \leq i \leq N$.
The *concatenation* of two strings $\mathbf{s} \in \mathbb{B}(N)$ and $\mathbf{t} \in \mathbb{B}(M)$ is

$$\mathbf{s} \cdot \mathbf{t} = s_1 s_2 \cdots s_N t_1 t_2 \cdots t_M = \mathbf{st}.$$

Exponentiation represents repeated concatenation of the same string.

$$\mathbf{s}^k = \underbrace{\mathbf{s} \cdot \mathbf{s} \cdots \mathbf{s}}_{\text{k copies}}$$

Binary strings of equal length are totally ordered by their numerical value. That is, if $\mathbf{s}, \mathbf{t} \in \mathbb{B}(N)$ then $\mathbf{s} < \mathbf{t}$ if there exists i such that

$$s_1 s_2 \cdots s_i = t_1 t_2 \cdot t_i \text{ and } s_{i+1} = 0 \text{ and } t_{i+1} = 1.$$

$\mathcal{L}(\mathbb{B})$ denotes this *lexicographic order* for any $\mathbb{B} \subseteq \mathbb{B}(N)$. For example,

$$\mathcal{L}(\mathbb{B}(3)) = 000, 001, 010, 011, 100, 101, 110, 111.$$

Concatenation and Lexicographic Order

If $\mathbf{s} \in \mathbb{B}(N)$ then s_i refers to the ith bit of $\mathbf{s}$ for $1 \leq i \leq N$.
The *concatenation* of two strings $\mathbf{s} \in \mathbb{B}(N)$ and $\mathbf{t} \in \mathbb{B}(M)$ is

$$\mathbf{s} \cdot \mathbf{t} = s_1 s_2 \cdots s_N t_1 t_2 \cdots t_M = \mathbf{st}.$$

Exponentiation represents repeated concatenation of the same string.

$$\mathbf{s}^k = \underbrace{\mathbf{s} \cdot \mathbf{s} \cdots \mathbf{s}}_{\text{k copies}}$$

Binary strings of equal length are totally ordered by their numerical value. That is, if $\mathbf{s}, \mathbf{t} \in \mathbb{B}(N)$ then $\mathbf{s} < \mathbf{t}$ if there exists i such that

$$s_1 s_2 \cdots s_i = t_1 t_2 \cdot t_i \text{ and } s_{i+1} = 0 \text{ and } t_{i+1} = 1.$$

$\mathcal{L}(\mathbb{B})$ denotes this *lexicographic order* for any $\mathbb{B} \subseteq \mathbb{B}(N)$. For example,

$$\mathcal{L}(\mathbb{B}(3)) = 000, 001, 010, 011, 100, 101, 110, 111.$$

Concatenation and Lexicographic Order

If $\mathbf{s} \in \mathbb{B}(N)$ then s_i refers to the ith bit of $\mathbf{s}$ for $1 \leq i \leq N$.
The *concatenation* of two strings $\mathbf{s} \in \mathbb{B}(N)$ and $\mathbf{t} \in \mathbb{B}(M)$ is

$$\mathbf{s} \cdot \mathbf{t} = s_1 s_2 \cdots s_N t_1 t_2 \cdots t_M = \mathbf{st}.$$

Exponentiation represents repeated concatenation of the same string.

$$\mathbf{s}^k = \underbrace{\mathbf{s} \cdot \mathbf{s} \cdots \mathbf{s}}_{\text{k copies}}$$

Binary strings of equal length are totally ordered by their numerical value. That is, if $\mathbf{s}, \mathbf{t} \in \mathbb{B}(N)$ then $\mathbf{s} < \mathbf{t}$ if there exists i such that

$$s_1 s_2 \cdots s_i = t_1 t_2 \cdot t_i \text{ and } s_{i+1} = 0 \text{ and } t_{i+1} = 1.$$

$\mathcal{L}(\mathbb{B})$ denotes this *lexicographic order* for any $\mathbb{B} \subseteq \mathbb{B}(N)$. For example,

$$\mathcal{L}(\mathbb{B}(3)) = 000, 001, 010, 011, 100, 101, 110, 111.$$

Concatenation and Lexicographic Order

If $\mathbf{s} \in \mathbb{B}(N)$ then s_i refers to the ith bit of $\mathbf{s}$ for $1 \leq i \leq N$.
The *concatenation* of two strings $\mathbf{s} \in \mathbb{B}(N)$ and $\mathbf{t} \in \mathbb{B}(M)$ is

$$\mathbf{s} \cdot \mathbf{t} = s_1 s_2 \cdots s_N t_1 t_2 \cdots t_M = \mathbf{st}.$$

Exponentiation represents repeated concatenation of the same string.

$$\mathbf{s}^k = \underbrace{\mathbf{s} \cdot \mathbf{s} \cdots \mathbf{s}}_{\text{k copies}}$$

Binary strings of equal length are totally ordered by their numerical value. That is, if $\mathbf{s}, \mathbf{t} \in \mathbb{B}(N)$ then $\mathbf{s} < \mathbf{t}$ if there exists i such that

$$s_1 s_2 \cdots s_i = t_1 t_2 \cdot t_i \text{ and } s_{i+1} = 0 \text{ and } t_{i+1} = 1.$$

$\mathcal{L}(\mathbb{B})$ denotes this *lexicographic order* for any $\mathbb{B} \subseteq \mathbb{B}(N)$. For example,

$$\mathcal{L}(\mathbb{B}(3)) = 000, 001, 010, 011, 100, 101, 110, 111.$$

Concatenation and Lexicographic Order

If $\mathbf{s} \in \mathbb{B}(N)$ then s_i refers to the ith bit of $\mathbf{s}$ for $1 \leq i \leq N$.
The *concatenation* of two strings $\mathbf{s} \in \mathbb{B}(N)$ and $\mathbf{t} \in \mathbb{B}(M)$ is

$$\mathbf{s} \cdot \mathbf{t} = s_1 s_2 \cdots s_N t_1 t_2 \cdots t_M = \mathbf{st}.$$

Exponentiation represents repeated concatenation of the same string.

$$\mathbf{s}^k = \underbrace{\mathbf{s} \cdot \mathbf{s} \cdots \mathbf{s}}_{\text{k copies}}$$

Binary strings of equal length are totally ordered by their numerical value. That is, if $\mathbf{s}, \mathbf{t} \in \mathbb{B}(N)$ then $\mathbf{s} < \mathbf{t}$ if there exists i such that

$$s_1 s_2 \cdots s_i = t_1 t_2 \cdot t_i \text{ and } s_{i+1} = 0 \text{ and } t_{i+1} = 1.$$

$\mathcal{L}(\mathbb{B})$ denotes this *lexicographic order* for any $\mathbb{B} \subseteq \mathbb{B}(N)$. For example,

$$\mathcal{L}(\mathbb{B}(3)) = 000, 001, 010, 011, 100, 101, 110, 111.$$

Concatenation and Lexicographic Order

If $\mathbf{s} \in \mathbb{B}(N)$ then s_i refers to the ith bit of $\mathbf{s}$ for $1 \leq i \leq N$.
The *concatenation* of two strings $\mathbf{s} \in \mathbb{B}(N)$ and $\mathbf{t} \in \mathbb{B}(M)$ is

$$\mathbf{s} \cdot \mathbf{t} = s_1 s_2 \cdots s_N t_1 t_2 \cdots t_M = \mathbf{st}.$$

Exponentiation represents repeated concatenation of the same string.

$$\mathbf{s}^k = \underbrace{\mathbf{s} \cdot \mathbf{s} \cdots \mathbf{s}}_{\text{k copies}}$$

Binary strings of equal length are totally ordered by their numerical value. That is, if $\mathbf{s}, \mathbf{t} \in \mathbb{B}(N)$ then $\mathbf{s} < \mathbf{t}$ if there exists i such that

$$s_1 s_2 \cdots s_i = t_1 t_2 \cdot t_i \text{ and } s_{i+1} = 0 \text{ and } t_{i+1} = 1.$$

$\mathcal{L}(\mathbb{B})$ denotes this *lexicographic order* for any $\mathbb{B} \subseteq \mathbb{B}(N)$. For example,

$$\mathcal{L}(\mathbb{B}(3)) = 000, 001, 010, 011, 100, 101, 110, 111.$$

Concatenation and Lexicographic Order

If $\mathbf{s} \in \mathbb{B}(N)$ then s_i refers to the ith bit of $\mathbf{s}$ for $1 \leq i \leq N$.
The *concatenation* of two strings $\mathbf{s} \in \mathbb{B}(N)$ and $\mathbf{t} \in \mathbb{B}(M)$ is

$$\mathbf{s} \cdot \mathbf{t} = s_1 s_2 \cdots s_N t_1 t_2 \cdots t_M = \mathbf{st}.$$

Exponentiation represents repeated concatenation of the same string.

$$\mathbf{s}^k = \underbrace{\mathbf{s} \cdot \mathbf{s} \cdots \mathbf{s}}_{\text{k copies}}$$

Binary strings of equal length are totally ordered by their numerical value. That is, if $\mathbf{s}, \mathbf{t} \in \mathbb{B}(N)$ then $\mathbf{s} < \mathbf{t}$ if there exists i such that

$$s_1 s_2 \cdots s_i = t_1 t_2 \cdot t_i \text{ and } s_{i+1} = 0 \text{ and } t_{i+1} = 1.$$

$\mathcal{L}(\mathbb{B})$ denotes this *lexicographic order* for any $\mathbb{B} \subseteq \mathbb{B}(N)$. For example,

$$\mathcal{L}(\mathbb{B}(3)) = 000, 001, 010, 011, 100, 101, 110, 111.$$

Rotations and Necklaces

The *ith rotation* of $\mathbf{s}$ is $s_i s_{i+1} \cdots s_N s_1 s_2 \cdot s_{i-1}$. The *rotation set* of $\mathbf{s}$ is

$$\circledcirc(\mathbf{s}) = \{s_i s_{i+1} \cdots s_N s_1 s_2 \cdots s_{i-1} \mid 1 \leq i \leq N\}.$$

Notice that $|\circledcirc(\mathbf{s})|$ divides N. For example,

$$\circledcirc(101010) = \{101010, 010101\}$$
$$\circledcirc(011011) = \{011011, 110110, 101101\}$$
$$\circledcirc(010001) = \{010001, 100010, 000101, 001010, 010001, 101000\}$$

A string $\mathbf{s}$ is a *necklace* if $\mathbf{s} \leq \mathbf{b} \cdot \mathbf{a}$ whenever $\mathbf{s} = \mathbf{a} \cdot \mathbf{b}$. In other words, $\mathbf{s}$ is a necklace if it is the lexicographically smallest string in its rotation set. For example, 000101 is a necklace and 010001 is not.

The set of necklaces of length N is $\mathbb{N}(N) \subseteq \mathbb{B}(N)$. For example,

$$\mathbb{N}(4) = \{0000, 0001, 0011, 0101, 0111, 1111\}.$$

Rotations and Necklaces

The *ith rotation* of $\mathbf{s}$ is $s_i s_{i+1} \cdots s_N s_1 s_2 \cdot s_{i-1}$. The *rotation set* of $\mathbf{s}$ is

$$\circledcirc(\mathbf{s}) = \{s_i s_{i+1} \cdots s_N s_1 s_2 \cdots s_{i-1} \mid 1 \leq i \leq N\}.$$

Notice that $|\circledcirc(\mathbf{s})|$ divides N. For example,

$$\circledcirc(101010) = \{101010, 010101\}$$
$$\circledcirc(011011) = \{011011, 110110, 101101\}$$
$$\circledcirc(010001) = \{010001, 100010, 000101, 001010, 010001, 101000\}$$

A string $\mathbf{s}$ is a *necklace* if $\mathbf{s} \leq \mathbf{b} \cdot \mathbf{a}$ whenever $\mathbf{s} = \mathbf{a} \cdot \mathbf{b}$. In other words, $\mathbf{s}$ is a necklace if it is the lexicographically smallest string in its rotation set. For example, 000101 is a necklace and 010001 is not.

The set of necklaces of length N is $\mathbb{N}(N) \subseteq \mathbb{B}(N)$. For example,

$$\mathbb{N}(4) = \{0000, 0001, 0011, 0101, 0111, 1111\}.$$

Rotations and Necklaces

The *ith rotation* of $\mathbf{s}$ is $s_i s_{i+1} \cdots s_N s_1 s_2 \cdot s_{i-1}$. The *rotation set* of $\mathbf{s}$ is

$$\circledcirc(\mathbf{s}) = \{s_i s_{i+1} \cdots s_N s_1 s_2 \cdots s_{i-1} \mid 1 \leq i \leq N\}.$$

Notice that $|\circledcirc(\mathbf{s})|$ divides N. For example,

$$\circledcirc(101010) = \{101010, 010101\}$$
$$\circledcirc(011011) = \{011011, 110110, 101101\}$$
$$\circledcirc(010001) = \{010001, 100010, 000101, 001010, 010001, 101000\}$$

A string $\mathbf{s}$ is a *necklace* if $\mathbf{s} \leq \mathbf{b} \cdot \mathbf{a}$ whenever $\mathbf{s} = \mathbf{a} \cdot \mathbf{b}$. In other words, $\mathbf{s}$ is a necklace if it is the lexicographically smallest string in its rotation set. For example, 000101 is a necklace and 010001 is not.

The set of necklaces of length N is $\mathbb{N}(N) \subseteq \mathbb{B}(N)$. For example,

$$\mathbb{N}(4) = \{0000, 0001, 0011, 0101, 0111, 1111\}.$$

Rotations and Necklaces

The *ith rotation* of $\mathbf{s}$ is $s_i s_{i+1} \cdots s_N s_1 s_2 \cdot s_{i-1}$. The *rotation set* of $\mathbf{s}$ is

$$\circledcirc(\mathbf{s}) = \{s_i s_{i+1} \cdots s_N s_1 s_2 \cdots s_{i-1} \mid 1 \leq i \leq N\}.$$

Notice that $|\circledcirc(\mathbf{s})|$ divides N. For example,

$$\circledcirc(101010) = \{101010, 010101\}$$
$$\circledcirc(011011) = \{011011, 110110, 101101\}$$
$$\circledcirc(010001) = \{010001, 100010, 000101, 001010, 010001, 101000\}$$

A string $\mathbf{s}$ is a *necklace* if $\mathbf{s} \leq \mathbf{b} \cdot \mathbf{a}$ whenever $\mathbf{s} = \mathbf{a} \cdot \mathbf{b}$. In other words, $\mathbf{s}$ is a necklace if it is the lexicographically smallest string in its rotation set. For example, 000101 is a necklace and 010001 is not.

The set of necklaces of length N is $\mathbb{N}(N) \subseteq \mathbb{B}(N)$. For example,

$$\mathbb{N}(4) = \{0000, 0001, 0011, 0101, 0111, 1111\}.$$

Rotations and Necklaces

The *ith rotation* of $\mathbf{s}$ is $s_i s_{i+1} \cdots s_N s_1 s_2 \cdot s_{i-1}$. The *rotation set* of $\mathbf{s}$ is

$$\circledcirc(\mathbf{s}) = \{s_i s_{i+1} \cdots s_N s_1 s_2 \cdots s_{i-1} \mid 1 \leq i \leq N\}.$$

Notice that $|\circledcirc(\mathbf{s})|$ divides N. For example,

$$\circledcirc(101010) = \{101010, 010101\}$$
$$\circledcirc(011011) = \{011011, 110110, 101101\}$$
$$\circledcirc(010001) = \{010001, 100010, 000101, 001010, 010001, 101000\}$$

A string $\mathbf{s}$ is a *necklace* if $\mathbf{s} \leq \mathbf{b} \cdot \mathbf{a}$ whenever $\mathbf{s} = \mathbf{a} \cdot \mathbf{b}$. In other words, $\mathbf{s}$ is a necklace if it is the lexicographically smallest string in its rotation set. For example, 000101 is a necklace and 010001 is not.

The set of necklaces of length N is $\mathbb{N}(N) \subseteq \mathbb{B}(N)$. For example,

$$\mathbb{N}(4) = \{0000, 0001, 0011, 0101, 0111, 1111\}.$$

Rotations and Necklaces

The *ith rotation* of $\mathbf{s}$ is $s_i s_{i+1} \cdots s_N s_1 s_2 \cdot s_{i-1}$. The *rotation set* of $\mathbf{s}$ is

$$\circledcirc(\mathbf{s}) = \{s_i s_{i+1} \cdots s_N s_1 s_2 \cdots s_{i-1} \mid 1 \leq i \leq N\}.$$

Notice that $|\circledcirc(\mathbf{s})|$ divides N. For example,

$$\circledcirc(101010) = \{101010, 010101\}$$
$$\circledcirc(011011) = \{011011, 110110, 101101\}$$
$$\circledcirc(010001) = \{010001, 100010, 000101, 001010, 010001, 101000\}$$

A string $\mathbf{s}$ is a *necklace* if $\mathbf{s} \leq \mathbf{b} \cdot \mathbf{a}$ whenever $\mathbf{s} = \mathbf{a} \cdot \mathbf{b}$. In other words, $\mathbf{s}$ is a necklace if it is the lexicographically smallest string in its rotation set. For example, 000101 is a necklace and 010001 is not.

The set of necklaces of length N is $\mathbb{N}(N) \subseteq \mathbb{B}(N)$. For example,

$$\mathbb{N}(4) = \{0000, 0001, 0011, 0101, 0111, 1111\}.$$

Rotations and Necklaces

The *ith rotation* of $\mathbf{s}$ is $s_i s_{i+1} \cdots s_N s_1 s_2 \cdot s_{i-1}$. The *rotation set* of $\mathbf{s}$ is

$$\circledcirc(\mathbf{s}) = \{s_i s_{i+1} \cdots s_N s_1 s_2 \cdots s_{i-1} \mid 1 \leq i \leq N\}.$$

Notice that $|\circledcirc(\mathbf{s})|$ divides N. For example,

$$\circledcirc(101010) = \{101010, 010101\}$$
$$\circledcirc(011011) = \{011011, 110110, 101101\}$$
$$\circledcirc(010001) = \{010001, 100010, 000101, 001010, 010001, 101000\}$$

A string $\mathbf{s}$ is a *necklace* if $\mathbf{s} \leq \mathbf{b} \cdot \mathbf{a}$ whenever $\mathbf{s} = \mathbf{a} \cdot \mathbf{b}$. In other words, $\mathbf{s}$ is a necklace if it is the lexicographically smallest string in its rotation set. For example, 000101 is a necklace and 010001 is not.

The set of necklaces of length N is $\mathbb{N}(N) \subseteq \mathbb{B}(N)$. For example,

$$\mathbb{N}(4) = \{0000, 0001, 0011, 0101, 0111, 1111\}.$$

Rotations and Necklaces

The *ith rotation* of $\mathbf{s}$ is $s_i s_{i+1} \cdots s_N s_1 s_2 \cdot s_{i-1}$. The *rotation set* of $\mathbf{s}$ is

$$\circledcirc(\mathbf{s}) = \{s_i s_{i+1} \cdots s_N s_1 s_2 \cdots s_{i-1} \mid 1 \leq i \leq N\}.$$

Notice that $|\circledcirc(\mathbf{s})|$ divides N. For example,

$$\circledcirc(101010) = \{101010, 010101\}$$
$$\circledcirc(011011) = \{011011, 110110, 101101\}$$
$$\circledcirc(010001) = \{010001, 100010, 000101, 001010, 010001, 101000\}$$

A string $\mathbf{s}$ is a *necklace* if $\mathbf{s} \leq \mathbf{b} \cdot \mathbf{a}$ whenever $\mathbf{s} = \mathbf{a} \cdot \mathbf{b}$. In other words, $\mathbf{s}$ is a necklace if it is the lexicographically smallest string in its rotation set. For example, 000101 is a necklace and 010001 is not.

The set of necklaces of length N is $\mathbb{N}(N) \subseteq \mathbb{B}(N)$. For example,

$$\mathbb{N}(4) = \{0000, 0001, 0011, 0101, 0111, 1111\}.$$

Rotations and Necklaces

The *ith rotation* of $\mathbf{s}$ is $s_i s_{i+1} \cdots s_N s_1 s_2 \cdot s_{i-1}$. The *rotation set* of $\mathbf{s}$ is

$$\circledcirc(\mathbf{s}) = \{s_i s_{i+1} \cdots s_N s_1 s_2 \cdots s_{i-1} \mid 1 \leq i \leq N\}.$$

Notice that $|\circledcirc(\mathbf{s})|$ divides N. For example,

$$\circledcirc(101010) = \{101010, 010101\}$$
$$\circledcirc(011011) = \{011011, 110110, 101101\}$$
$$\circledcirc(010001) = \{010001, 100010, 000101, 001010, 010001, 101000\}$$

A string $\mathbf{s}$ is a *necklace* if $\mathbf{s} \leq \mathbf{b} \cdot \mathbf{a}$ whenever $\mathbf{s} = \mathbf{a} \cdot \mathbf{b}$. In other words, $\mathbf{s}$ is a necklace if it is the lexicographically smallest string in its rotation set. For example, 000101 is a necklace and 010001 is not.

The set of necklaces of length N is $\mathbb{N}(N) \subseteq \mathbb{B}(N)$. For example,

$$\mathbb{N}(4) = \{0000, 0001, 0011, 0101, 0111, 1111\}.$$

Rotations and Necklaces

The *ith rotation* of $\mathbf{s}$ is $s_i s_{i+1} \cdots s_N s_1 s_2 \cdot s_{i-1}$. The *rotation set* of $\mathbf{s}$ is

$$\circledcirc(\mathbf{s}) = \{s_i s_{i+1} \cdots s_N s_1 s_2 \cdots s_{i-1} \mid 1 \leq i \leq N\}.$$

Notice that $|\circledcirc(\mathbf{s})|$ divides N. For example,

$$\circledcirc(101010) = \{101010, 010101\}$$
$$\circledcirc(011011) = \{011011, 110110, 101101\}$$
$$\circledcirc(010001) = \{010001, 100010, 000101, 001010, 010001, 101000\}$$

A string $\mathbf{s}$ is a *necklace* if $\mathbf{s} \leq \mathbf{b} \cdot \mathbf{a}$ whenever $\mathbf{s} = \mathbf{a} \cdot \mathbf{b}$. In other words, $\mathbf{s}$ is a necklace if it is the lexicographically smallest string in its rotation set. For example, 000101 is a necklace and 010001 is not.

The set of necklaces of length N is $\mathbb{N}(N) \subseteq \mathbb{B}(N)$. For example,

$$\mathbb{N}(4) = \{0000, 0001, 0011, 0101, 0111, 1111\}.$$

Rotations and Necklaces

The *ith rotation* of $\mathbf{s}$ is $s_i s_{i+1} \cdots s_N s_1 s_2 \cdot s_{i-1}$. The *rotation set* of $\mathbf{s}$ is

$$\circledcirc(\mathbf{s}) = \{s_i s_{i+1} \cdots s_N s_1 s_2 \cdots s_{i-1} \mid 1 \leq i \leq N\}.$$

Notice that $|\circledcirc(\mathbf{s})|$ divides N. For example,

$$\circledcirc(101010) = \{101010, 010101\}$$
$$\circledcirc(011011) = \{011011, 110110, 101101\}$$
$$\circledcirc(010001) = \{010001, 100010, 000101, 001010, 010001, 101000\}$$

A string $\mathbf{s}$ is a *necklace* if $\mathbf{s} \leq \mathbf{b} \cdot \mathbf{a}$ whenever $\mathbf{s} = \mathbf{a} \cdot \mathbf{b}$. In other words, $\mathbf{s}$ is a necklace if it is the lexicographically smallest string in its rotation set. For example, 000101 is a necklace and 010001 is not.

The set of necklaces of length N is $\mathbb{N}(N) \subseteq \mathbb{B}(N)$. For example,

$$\mathbb{N}(4) = \{0000, 0001, 0011, 0101, 0111, 1111\}.$$

Prime-Prefixes

The *prime-prefix* of $\mathbf{s}$ is

$$\text{prime}(\mathbf{s}) = \mathbf{p}$$

where $\mathbf{p}$ is the shortest prefix of $\mathbf{s}$ such that $\mathbf{s} = \mathbf{p}^k$ for some $k > 0$.

For example,

$$\text{prime}(101010) = 10$$
$$\text{prime}(011011) = 011$$
$$\text{prime}(010001) = 010001.$$

Notice that

$$|\text{prime}(\mathbf{s})| = |\circledcirc(\mathbf{s})|.$$

A string is *prime* if $\text{prime}(\mathbf{s}) = \mathbf{s}$ and prime necklaces are also known as *Lyndon words*.

Prime-Prefixes

The *prime-prefix* of $\mathbf{s}$ is

$$\text{prime}(\mathbf{s}) = \mathbf{p}$$

where $\mathbf{p}$ is the shortest prefix of $\mathbf{s}$ such that $\mathbf{s} = \mathbf{p}^k$ for some $k > 0$. For example,

$$\text{prime}(101010) = 10$$
$$\text{prime}(011011) = 011$$
$$\text{prime}(010001) = 010001.$$

Notice that

$$|\text{prime}(\mathbf{s})| = |\circledcirc(\mathbf{s})|.$$

A string is *prime* if $\text{prime}(\mathbf{s}) = \mathbf{s}$ and prime necklaces are also known as *Lyndon words*.

Prime-Prefixes

The *prime-prefix* of $\mathbf{s}$ is

$$\text{prime}(\mathbf{s}) = \mathbf{p}$$

where $\mathbf{p}$ is the shortest prefix of $\mathbf{s}$ such that $\mathbf{s} = \mathbf{p}^k$ for some $k > 0$.
For example,

$$\text{prime}(101010) = 10$$
$$\text{prime}(011011) = 011$$
$$\text{prime}(010001) = 010001.$$

Notice that

$$|\text{prime}(\mathbf{s})| = |\circledcirc(\mathbf{s})|.$$

A string is *prime* if $\text{prime}(\mathbf{s}) = \mathbf{s}$ and prime necklaces are also known as *Lyndon words*.

Prime-Prefixes

The *prime-prefix* of $\mathbf{s}$ is

$$\text{prime}(\mathbf{s}) = \mathbf{p}$$

where $\mathbf{p}$ is the shortest prefix of $\mathbf{s}$ such that $\mathbf{s} = \mathbf{p}^k$ for some $k > 0$.
For example,

$$\text{prime}(101010) = 10$$
$$\text{prime}(011011) = 011$$
$$\text{prime}(010001) = 010001.$$

Notice that

$$|\text{prime}(\mathbf{s})| = |\circledcirc(\mathbf{s})|.$$

A string is *prime* if $\text{prime}(\mathbf{s}) = \mathbf{s}$ and prime necklaces are also known as *Lyndon words*.

Prime-Prefixes

The *prime-prefix* of $\mathbf{s}$ is

$$\text{prime}(\mathbf{s}) = \mathbf{p}$$

where $\mathbf{p}$ is the shortest prefix of $\mathbf{s}$ such that $\mathbf{s} = \mathbf{p}^k$ for some $k > 0$.
For example,

$$\begin{aligned}\text{prime}(101010) &= 10\\ \text{prime}(011011) &= 011\\ \text{prime}(010001) &= 010001.\end{aligned}$$

Notice that

$$|\text{prime}(\mathbf{s})| = |\circledcirc(\mathbf{s})|.$$

A string is *prime* if $\text{prime}(\mathbf{s}) = \mathbf{s}$ and prime necklaces are also known as *Lyndon words*.

Construction using Lexicographic Order

$\mathcal{L}(\mathbb{B}(4))$	$\mathcal{L}(\mathbb{N}(4))$	prime	$\cdot$	$\mathbb{B}(4)$
0000	0000	0	0	0000
0001	0001	0001	0	0001
0010			0	0010
0011	0011	0011	0	0100
0100			1	1001
0101	0101	01	0	0011
0110			0	0110
0111	0111	0111	1	1101
1000			1	1010
1001			0	0101
1010			1	1011
1011			0	0111
1100			1	1111
1101			1	1110
1110			1	1100
1111	1111	1	1	1000

This construction creates a de Bruijn cycle for any $N \geq 0$.

Construction using Lexicographic Order

$\mathcal{L}(\mathbb{B}(4))$	$\mathcal{L}(\mathbb{N}(4))$	prime	$\cdot$	$\mathbb{B}(4)$
0000	0000	0	0	0000
0001	0001	0001	0	0001
0010			0	0010
0011	0011	0011	0	0100
0100			1	1001
0101	0101	01	0	0011
0110			0	0110
0111	0111	0111	1	1101
1000			1	1010
1001			0	0101
1010			1	1011
1011			0	0111
1100			1	1111
1101			1	1110
1110			1	1100
1111	1111	1	1	1000

1. Start with the lexicographic order of binary strings of length N.

Construction using Lexicographic Order

$\mathcal{L}(\mathbb{B}(4))$	$\mathcal{L}(\mathbb{N}(4))$	prime	$\cdot$	$\mathbb{B}(4)$
0000	0000	0	0	0000
0001	0001	0001	0	0001
0010			0	0010
0011	0011	0011	0	0100
0100			1	1001
0101	0101	01	0	0011
0110			0	0110
0111	0111	0111	1	1101
1000			1	1010
1001			0	0101
1010			1	1011
1011			0	0111
1100			1	1111
1101			1	1110
1110			1	1100
1111	1111	1	1	1000

2. Remove non-necklaces to get lexicographic order of necklaces of length N.

Construction using Lexicographic Order

$\mathcal{L}(\mathbb{B}(4))$	$\mathcal{L}(\mathbb{N}(4))$	prime	$\cdot$	$\mathbb{B}(4)$
0000	0000	0	0	0000
0001	0001	0001	0	0001
0010			0	0010
0011	0011	0011	0	0100
0100			1	1001
0101	0101	01	0	0011
0110			0	0110
0111	0111	0111	1	1101
1000			1	1010
1001			0	0101
1010			1	1011
1011			0	0111
1100			1	1111
1101			1	1110
1110			1	1100
1111	1111	1	1	1000

3. Take the prime-prefix of each necklace.

Construction using Lexicographic Order

$\mathcal{L}(\mathbb{B}(4))$	$\mathcal{L}(\mathbb{N}(4))$	prime	$\cdot$	$\mathbb{B}(4)$
0000	0000	0	0	0000
0001	0001	0001	0	0001
0010			0	0010
0011	0011	0011	0	0100
0100			1	1001
0101	0101	01	0	0011
0110			0	0110
0111	0111	0111	1	1101
1000			1	1010
1001			0	0101
1010			1	1011
1011			0	0111
1100			1	1111
1101			1	1110
1110			1	1100
1111	1111	1	1	1000

4. Concatenate the results to obtain the de Bruijn cycle.

Construction using Lexicographic Order

$\mathcal{L}(\mathbb{B}(4))$	$\mathcal{L}(\mathbb{N}(4))$	prime	$\cdot$	$\mathbb{B}(4)$
0000	0000	0	0	0000
0001	0001	0001	0	0001
0010			0	0010
0011	0011	0011	0	0100
0100			1	1001
0101	0101	01	0	0011
0110			0	0110
0111	0111	0111	1	1101
1000			1	1010
1001			0	0101
1010			1	1011
1011			0	0111
1100			1	1111
1101			1	1110
1110			1	1100
1111	1111	1	1	1000

(Correct length since $\circledcirc(\mathbf{s})$ contributes $|\circledcirc(\mathbf{s})|$ bits for each $\mathbf{s} \in \mathbb{B}(N)$.)

Construction using Lexicographic Order

$\mathcal{L}(\mathbb{B}(4))$	$\mathcal{L}(\mathbb{N}(4))$	prime	$\cdot$	$\mathbb{B}(4)$
0000	0000	0	0	0000
0001	0001	0001	0	0001
0010			0	0010
0011	0011	0011	0	0100
0100			1	1001
0101	0101	01	0	0011
0110			0	0110
0111	0111	0111	1	1101
1000			1	1010
1001			0	0101
1010			1	1011
1011			0	0111
1100			1	1111
1101			1	1110
1110			1	1100
1111	1111	1	1	1000

5. Knuth calls this lexicographically smallest de Bruijn cycle the *Grand-Daddy*.

Construction using Lexicographic Order

$\mathcal{L}(\mathbb{B}(4))$	$\mathcal{L}(\mathbb{N}(4))$	prime	$\cdot$	$\mathbb{B}(4)$
0000	0000	0	0	0000
0001	0001	0001	0	0001
0010			0	0010
0011	0011	0011	0	0100
0100			1	1001
0101	0101	01	0	0011
0110			0	0110
0111	0111	0111	1	1101
1000			1	1010
1001			0	0101
1010			1	1011
1011			0	0111
1100			1	1111
1101			1	1110
1110			1	1100
1111	1111	1	1	1000

[Martin 1934, Fredericksen-Maiorana 1978, Fredericksen-Kessler 1986, Ruskey-Savage-Wang 1992]

Fixed-Density Binary Strings

Let $\mathbb{B}(N, D)$ be the set of length N binary strings with *density* D.

$$|\mathbb{B}(N, D)| = \binom{D}{N}.$$

The strings in $\mathbb{B}(N, D)$ are *fixed-density binary strings*. (Similarly, the set of *fixed-density necklaces* is denoted $\mathbb{N}(N, D) \subseteq \mathbb{B}(N, D)$.)

Strings of length $\binom{N}{D}$ containing every string in $\mathbb{B}(N, D)$ exactly once as a (circular) substring do not exist unless $\min(D, N - D) \leq 1$. For example, wlog the following would be such a string for $N = 4$ and $D = 2$

$$0011xy$$

However, its substrings of length 4 are

$$0011, 011x, 11xy, 1xy0, xy00, y001$$

and $11xy$ and $xy00$ cannot both be in $\mathbb{B}(4, 2)$.

Idea: The last bit of each string in $\mathbb{B}(N, D)$ is *redundant*. For example, if $010x \in \mathbb{B}(4, 2)$ then $x = 1$. Thus, the Nth bit can be omitted and then inferred from the first $N - 1$ bits.

Fixed-Density Binary Strings

Let $\mathbb{B}(N, D)$ be the set of length N binary strings with *density* D.

$$|\mathbb{B}(N, D)| = \binom{D}{N}.$$

The strings in $\mathbb{B}(N, D)$ are *fixed-density binary strings*. (Similarly, the set of *fixed-density necklaces* is denoted $\mathbb{N}(N, D) \subseteq \mathbb{B}(N, D)$.)

Strings of length $\binom{N}{D}$ containing every string in $\mathbb{B}(N, D)$ exactly once as a (circular) substring do not exist unless $\min(D, N - D) \leq 1$. For example, wlog the following would be such a string for $N = 4$ and $D = 2$

$$0011xy$$

However, its substrings of length 4 are

$$0011, 011x, 11xy, 1xy0, xy00, y001$$

and $11xy$ and $xy00$ cannot both be in $\mathbb{B}(4, 2)$.

Idea: The last bit of each string in $\mathbb{B}(N, D)$ is *redundant*. For example, if $010x \in \mathbb{B}(4, 2)$ then $x = 1$. Thus, the Nth bit can be omitted and then inferred from the first $N - 1$ bits.

Fixed-Density Binary Strings

Let $\mathbb{B}(N, D)$ be the set of length N binary strings with *density* D.

$$|\mathbb{B}(N, D)| = \binom{D}{N}.$$

The strings in $\mathbb{B}(N, D)$ are *fixed-density binary strings*. (Similarly, the set of *fixed-density necklaces* is denoted $\mathbb{N}(N, D) \subseteq \mathbb{B}(N, D)$.)

Strings of length $\binom{N}{D}$ containing every string in $\mathbb{B}(N, D)$ exactly once as a (circular) substring do not exist unless $\min(D, N - D) \leq 1$. For example, wlog the following would be such a string for $N = 4$ and $D = 2$

$$0011xy$$

However, its substrings of length 4 are

$$0011, 011x, 11xy, 1xy0, xy00, y001$$

and $11xy$ and $xy00$ cannot both be in $\mathbb{B}(4, 2)$.

Idea: The last bit of each string in $\mathbb{B}(N, D)$ is *redundant*. For example, if $010x \in \mathbb{B}(4, 2)$ then $x = 1$. Thus, the Nth bit can be omitted and then inferred from the first $N - 1$ bits.

Fixed-Density Binary Strings

Let $\mathbb{B}(N, D)$ be the set of length N binary strings with *density* D.

$$|\mathbb{B}(N, D)| = \binom{D}{N}.$$

The strings in $\mathbb{B}(N, D)$ are *fixed-density binary strings*. (Similarly, the set of *fixed-density necklaces* is denoted $\mathbb{N}(N, D) \subseteq \mathbb{B}(N, D)$.)

Strings of length $\binom{N}{D}$ containing every string in $\mathbb{B}(N, D)$ exactly once as a (circular) substring do not exist unless $\min(D, N - D) \leq 1$. For example, wlog the following would be such a string for $N = 4$ and $D = 2$

$$0011xy$$

However, its substrings of length 4 are

$$0011, 011x, 11xy, 1xy0, xy00, y001$$

and $11xy$ and $xy00$ cannot both be in $\mathbb{B}(4, 2)$.

Idea: The last bit of each string in $\mathbb{B}(N, D)$ is *redundant*. For example, if $010x \in \mathbb{B}(4, 2)$ then $x = 1$. Thus, the Nth bit can be omitted and then inferred from the first $N - 1$ bits.

Fixed-Density Binary Strings

Let $\mathbb{B}(N, D)$ be the set of length N binary strings with *density* D.

$$|\mathbb{B}(N, D)| = \binom{D}{N}.$$

The strings in $\mathbb{B}(N, D)$ are *fixed-density binary strings*. (Similarly, the set of *fixed-density necklaces* is denoted $\mathbb{N}(N, D) \subseteq \mathbb{B}(N, D)$.)

Strings of length $\binom{N}{D}$ containing every string in $\mathbb{B}(N, D)$ exactly once as a (circular) substring do not exist unless $\min(D, N - D) \leq 1$. For example, wlog the following would be such a string for $N = 4$ and $D = 2$

$$0011xy$$

However, its substrings of length 4 are

$$0011, 011x, 11xy, 1xy0, xy00, y001$$

and $11xy$ and $xy00$ cannot both be in $\mathbb{B}(4, 2)$.

Idea: The last bit of each string in $\mathbb{B}(N, D)$ is *redundant*. For example, if $010x \in \mathbb{B}(4, 2)$ then $x = 1$. Thus, the Nth bit can be omitted and then inferred from the first $N - 1$ bits.

Fixed-Density Binary Strings

Let $\mathbb{B}(N, D)$ be the set of length N binary strings with *density* D.

$$|\mathbb{B}(N, D)| = \binom{D}{N}.$$

The strings in $\mathbb{B}(N, D)$ are *fixed-density binary strings*. (Similarly, the set of *fixed-density necklaces* is denoted $\mathbb{N}(N, D) \subseteq \mathbb{B}(N, D)$.)

Strings of length $\binom{N}{D}$ containing every string in $\mathbb{B}(N, D)$ exactly once as a (circular) substring do not exist unless $\min(D, N - D) \leq 1$. For example, wlog the following would be such a string for $N = 4$ and $D = 2$

$$0011xy$$

However, its substrings of length 4 are

$$0011, 011x, 11xy, 1xy0, xy00, y001$$

and $11xy$ and $xy00$ cannot both be in $\mathbb{B}(4, 2)$.

Idea: The last bit of each string in $\mathbb{B}(N, D)$ is *redundant*. For example, if $010x \in \mathbb{B}(4, 2)$ then $x = 1$. Thus, the Nth bit can be omitted and then inferred from the first $N - 1$ bits.

Fixed-Density Binary Strings

Let $\mathbb{B}(N, D)$ be the set of length N binary strings with *density* D.

$$|\mathbb{B}(N, D)| = \binom{D}{N}.$$

The strings in $\mathbb{B}(N, D)$ are *fixed-density binary strings*. (Similarly, the set of *fixed-density necklaces* is denoted $\mathbb{N}(N, D) \subseteq \mathbb{B}(N, D)$.)

Strings of length $\binom{N}{D}$ containing every string in $\mathbb{B}(N, D)$ exactly once as a (circular) substring do not exist unless $\min(D, N - D) \leq 1$. For example, wlog the following would be such a string for $N = 4$ and $D = 2$

$$0011xy$$

However, its substrings of length 4 are

$$0011, 011x, 11xy, 1xy0, xy00, y001$$

and $11xy$ and $xy00$ cannot both be in $\mathbb{B}(4, 2)$.

Idea: The last bit of each string in $\mathbb{B}(N, D)$ is *redundant*. For example, if $010x \in \mathbb{B}(4, 2)$ then $x = 1$. Thus, the Nth bit can be omitted and then inferred from the first $N - 1$ bits.

Fixed-Density Binary Strings

Let $\mathbb{B}(N, D)$ be the set of length N binary strings with *density* D.

$$|\mathbb{B}(N, D)| = \binom{D}{N}.$$

The strings in $\mathbb{B}(N, D)$ are *fixed-density binary strings*. (Similarly, the set of *fixed-density necklaces* is denoted $\mathbb{N}(N, D) \subseteq \mathbb{B}(N, D)$.)

Strings of length $\binom{N}{D}$ containing every string in $\mathbb{B}(N, D)$ exactly once as a (circular) substring do not exist unless $\min(D, N - D) \leq 1$. For example, wlog the following would be such a string for $N = 4$ and $D = 2$

$$0011xy$$

However, its substrings of length 4 are

$$0011, 011x, 11xy, 1xy0, xy00, y001$$

and $11xy$ and $xy00$ cannot both be in $\mathbb{B}(4, 2)$.

Idea: The last bit of each string in $\mathbb{B}(N, D)$ is *redundant*. For example, if $010x \in \mathbb{B}(4, 2)$ then $x = 1$. Thus, the Nth bit can be omitted and then inferred from the first $N - 1$ bits.

Fixed-Density Binary Strings

Let $\mathbb{B}(N, D)$ be the set of length N binary strings with *density* D.

$$|\mathbb{B}(N, D)| = \binom{D}{N}.$$

The strings in $\mathbb{B}(N, D)$ are *fixed-density binary strings*. (Similarly, the set of *fixed-density necklaces* is denoted $\mathbb{N}(N, D) \subseteq \mathbb{B}(N, D)$.)

Strings of length $\binom{N}{D}$ containing every string in $\mathbb{B}(N, D)$ exactly once as a (circular) substring do not exist unless $\min(D, N - D) \leq 1$. For example, wlog the following would be such a string for $N = 4$ and $D = 2$

$$0011xy$$

However, its substrings of length 4 are

$$0011, 011x, 11xy, 1xy0, xy00, y001$$

and $11xy$ and $xy00$ cannot both be in $\mathbb{B}(4, 2)$.

Idea: The last bit of each string in $\mathbb{B}(N, D)$ is *redundant*. For example, if $010x \in \mathbb{B}(4, 2)$ then $x = 1$. Thus, the Nth bit can be omitted and then inferred from the first $N - 1$ bits.

Fixed-Density Binary Strings

Let $\mathbb{B}(N, D)$ be the set of length N binary strings with *density* D.

$$|\mathbb{B}(N, D)| = \binom{D}{N}.$$

The strings in $\mathbb{B}(N, D)$ are *fixed-density binary strings*. (Similarly, the set of *fixed-density necklaces* is denoted $\mathbb{N}(N, D) \subseteq \mathbb{B}(N, D)$.)

Strings of length $\binom{N}{D}$ containing every string in $\mathbb{B}(N, D)$ exactly once as a (circular) substring do not exist unless $\min(D, N - D) \leq 1$. For example, wlog the following would be such a string for $N = 4$ and $D = 2$

$$0011xy$$

However, its substrings of length 4 are

$$0011, 011x, 11xy, 1xy0, xy00, y001$$

and $11xy$ and $xy00$ cannot both be in $\mathbb{B}(4, 2)$.

Idea: The last bit of each string in $\mathbb{B}(N, D)$ is *redundant*. For example, if $010x \in \mathbb{B}(4, 2)$ then $x = 1$. Thus, the Nth bit can be omitted and then inferred from the first $N - 1$ bits.

Fixed-Density Binary Strings

Let $\mathbb{B}(N, D)$ be the set of length N binary strings with *density* D.

$$|\mathbb{B}(N, D)| = \binom{D}{N}.$$

The strings in $\mathbb{B}(N, D)$ are *fixed-density binary strings*. (Similarly, the set of *fixed-density necklaces* is denoted $\mathbb{N}(N, D) \subseteq \mathbb{B}(N, D)$.)

Strings of length $\binom{N}{D}$ containing every string in $\mathbb{B}(N, D)$ exactly once as a (circular) substring do not exist unless $\min(D, N - D) \leq 1$. For example, wlog the following would be such a string for $N = 4$ and $D = 2$

$$0011xy$$

However, its substrings of length 4 are

$$0011, 011x, 11xy, 1xy0, xy00, y001$$

and $11xy$ and $xy00$ cannot both be in $\mathbb{B}(4, 2)$.

Idea: The last bit of each string in $\mathbb{B}(N, D)$ is *redundant*. For example, if $010x \in \mathbb{B}(4, 2)$ then $x = 1$. Thus, the Nth bit can be omitted and then inferred from the first $N - 1$ bits.

Fixed-Density Binary Strings

Let $\mathbb{B}(N, D)$ be the set of length N binary strings with *density* D.

$$|\mathbb{B}(N, D)| = \binom{D}{N}.$$

The strings in $\mathbb{B}(N, D)$ are *fixed-density binary strings*. (Similarly, the set of *fixed-density necklaces* is denoted $\mathbb{N}(N, D) \subseteq \mathbb{B}(N, D)$.)

Strings of length $\binom{N}{D}$ containing every string in $\mathbb{B}(N, D)$ exactly once as a (circular) substring do not exist unless $\min(D, N - D) \leq 1$. For example, wlog the following would be such a string for $N = 4$ and $D = 2$

$$0011xy$$

However, its substrings of length 4 are

$$0011, 011x, 11xy, 1xy0, xy00, y001$$

and $11xy$ and $xy00$ cannot both be in $\mathbb{B}(4, 2)$.

Idea: The last bit of each string in $\mathbb{B}(N, D)$ is *redundant*. For example, if $010x \in \mathbb{B}(4, 2)$ then $x = 1$. Thus, the Nth bit can be omitted and then inferred from the first $N - 1$ bits.

Fixed-Density Binary Strings

Let $\mathbb{B}(N, D)$ be the set of length N binary strings with *density* D.

$$|\mathbb{B}(N, D)| = \binom{D}{N}.$$

The strings in $\mathbb{B}(N, D)$ are *fixed-density binary strings*. (Similarly, the set of *fixed-density necklaces* is denoted $\mathbb{N}(N, D) \subseteq \mathbb{B}(N, D)$.)

Strings of length $\binom{N}{D}$ containing every string in $\mathbb{B}(N, D)$ exactly once as a (circular) substring do not exist unless $\min(D, N - D) \leq 1$. For example, wlog the following would be such a string for $N = 4$ and $D = 2$

$$0011xy$$

However, its substrings of length 4 are

$$0011, 011x, 11xy, 1xy0, xy00, y001$$

and $11xy$ and $xy00$ cannot both be in $\mathbb{B}(4, 2)$.

Idea: The last bit of each string in $\mathbb{B}(N, D)$ is *redundant*. For example, if $010x \in \mathbb{B}(4, 2)$ then $x = 1$. Thus, the Nth bit can be omitted and then inferred from the first $N - 1$ bits.

Fixed-Density Binary Strings

Let $\mathbb{B}(N, D)$ be the set of length N binary strings with *density* D.

$$|\mathbb{B}(N, D)| = \binom{D}{N}.$$

The strings in $\mathbb{B}(N, D)$ are *fixed-density binary strings*. (Similarly, the set of *fixed-density necklaces* is denoted $\mathbb{N}(N, D) \subseteq \mathbb{B}(N, D)$.)

Strings of length $\binom{N}{D}$ containing every string in $\mathbb{B}(N, D)$ exactly once as a (circular) substring do not exist unless $\min(D, N - D) \leq 1$. For example, wlog the following would be such a string for $N = 4$ and $D = 2$

$$0011xy$$

However, its substrings of length 4 are

$$0011, 011x, 11xy, 1xy0, xy00, y001$$

and $11xy$ and $xy00$ cannot both be in $\mathbb{B}(4, 2)$.

Idea: The last bit of each string in $\mathbb{B}(N, D)$ is *redundant*. For example, if $010x \in \mathbb{B}(4, 2)$ then $x = 1$. Thus, the Nth bit can be omitted and then inferred from the first $N - 1$ bits.

Fixed-Density Binary Strings

Let $\mathbb{B}(N, D)$ be the set of length N binary strings with *density* D.

$$|\mathbb{B}(N, D)| = \binom{D}{N}.$$

The strings in $\mathbb{B}(N, D)$ are *fixed-density binary strings*. (Similarly, the set of *fixed-density necklaces* is denoted $\mathbb{N}(N, D) \subseteq \mathbb{B}(N, D)$.)

Strings of length $\binom{N}{D}$ containing every string in $\mathbb{B}(N, D)$ exactly once as a (circular) substring do not exist unless $\min(D, N - D) \leq 1$. For example, wlog the following would be such a string for $N = 4$ and $D = 2$

$$0011xy$$

However, its substrings of length 4 are

$$0011, 011x, 11xy, 1xy0, xy00, y001$$

and $11xy$ and $xy00$ cannot both be in $\mathbb{B}(4, 2)$.

Idea: The last bit of each string in $\mathbb{B}(N, D)$ is *redundant*. For example, if $010x \in \mathbb{B}(4, 2)$ then $x = 1$. Thus, the Nth bit can be omitted and then inferred from the first $N - 1$ bits.

Fixed-Density Binary Strings

Let $\mathbb{B}(N, D)$ be the set of length N binary strings with *density* D.

$$|\mathbb{B}(N, D)| = \binom{D}{N}.$$

The strings in $\mathbb{B}(N, D)$ are *fixed-density binary strings*. (Similarly, the set of *fixed-density necklaces* is denoted $\mathbb{N}(N, D) \subseteq \mathbb{B}(N, D)$.)

Strings of length $\binom{N}{D}$ containing every string in $\mathbb{B}(N, D)$ exactly once as a (circular) substring do not exist unless $\min(D, N - D) \leq 1$. For example, wlog the following would be such a string for $N = 4$ and $D = 2$

$$0011xy$$

However, its substrings of length 4 are

$$0011, 011x, 11xy, 1xy0, xy00, y001$$

and $11xy$ and $xy00$ cannot both be in $\mathbb{B}(4, 2)$.

Idea: The last bit of each string in $\mathbb{B}(N, D)$ is *redundant*. For example, if $010x \in \mathbb{B}(4, 2)$ then $x = 1$. Thus, the Nth bit can be omitted and then inferred from the first $N - 1$ bits.

Fixed-Density Binary Strings

Let $\mathbb{B}(N, D)$ be the set of length N binary strings with *density* D.

$$|\mathbb{B}(N, D)| = \binom{D}{N}.$$

The strings in $\mathbb{B}(N, D)$ are *fixed-density binary strings*. (Similarly, the set of *fixed-density necklaces* is denoted $\mathbb{N}(N, D) \subseteq \mathbb{B}(N, D)$.)

Strings of length $\binom{N}{D}$ containing every string in $\mathbb{B}(N, D)$ exactly once as a (circular) substring do not exist unless $\min(D, N - D) \leq 1$. For example, wlog the following would be such a string for $N = 4$ and $D = 2$

$$0011xy$$

However, its substrings of length 4 are

$$0011, 011x, 11xy, 1xy0, xy00, y001$$

and $11xy$ and $xy00$ cannot both be in $\mathbb{B}(4, 2)$.

Idea: The last bit of each string in $\mathbb{B}(N, D)$ is *redundant*. For example, if $010x \in \mathbb{B}(4, 2)$ then $x = 1$. Thus, the Nth bit can be omitted and then inferred from the first $N - 1$ bits.

Fixed-Density de Bruijn Cycles

Let $\mathbb{B}'(N, D) = \mathbb{B}(N-1, D) \cup \mathbb{B}(N-1, D-1)$.

$$|\mathbb{B}'(N, D)| = |\mathbb{B}(N-1, D)| + |\mathbb{B}(N-1, D-1)| = \binom{N}{D}.$$

In particular, $\mathbb{B}'(2D, D)$ contains the *middle levels*. Strings in $\mathbb{B}'(N, D)$ and $\mathbb{B}(N, D)$ are *shorthand* and *longhand representations* of one another.

A *fixed-density de Bruijn cycle* is a length $\binom{N}{D}$ string containing every string in $\mathbb{B}'(N, D)$ exactly once as a (circular) substring. For example,

$$00011100110101001011$$

is a fixed-density de Bruijn cycle for $N = 6$ and $D = 3$ from its substrings

$$000111, 001110, 011100, 111000, 110010, 100110, 001101, 011010, 110100, 101010,$$
$$010101, 101001, 010011, 100101, 001011, 010110, 101101, 011001, 110001, 100011$$

Successive longhand strings differ by length $N-1$ or N *prefix right-shifts*. That is, if $\mathbf{t}$ follows $\mathbf{s}$ then $\mathbf{t} = \overrightarrow{\text{shift}}(\mathbf{s}, 1, j)$ for $j \in \{N-1, N\}$.

Question

Do fixed-density de Bruijn cycles exist for all N and D?

Fixed-Density de Bruijn Cycles

Let $\mathbb{B}'(N, D) = \mathbb{B}(N-1, D) \cup \mathbb{B}(N-1, D-1)$.

$$|\mathbb{B}'(N, D)| = |\mathbb{B}(N-1, D)| + |\mathbb{B}(N-1, D-1)| = \binom{N}{D}.$$

In particular, $\mathbb{B}'(2D, D)$ contains the *middle levels*. Strings in $\mathbb{B}'(N, D)$ and $\mathbb{B}(N, D)$ are *shorthand* and *longhand representations* of one another.

A *fixed-density de Bruijn cycle* is a length $\binom{N}{D}$ string containing every string in $\mathbb{B}'(N, D)$ exactly once as a (circular) substring. For example,

$$00011100110101001011$$

is a fixed-density de Bruijn cycle for $N = 6$ and $D = 3$ from its substrings

$$000111, 001110, 011100, 111000, 110010, 100110, 001101, 011010, 110100, 101010,$$
$$010101, 101001, 010011, 100101, 001011, 010110, 101101, 011001, 110001, 100011$$

Successive longhand strings differ by length $N-1$ or N *prefix right-shifts*. That is, if $\mathbf{t}$ follows $\mathbf{s}$ then $\mathbf{t} = \overrightarrow{\text{shift}}(\mathbf{s}, 1, j)$ for $j \in \{N-1, N\}$.

Question

Do fixed-density de Bruijn cycles exist for all N and D?

Fixed-Density de Bruijn Cycles

Let $\mathbb{B}'(N, D) = \mathbb{B}(N-1, D) \cup \mathbb{B}(N-1, D-1)$.

$$|\mathbb{B}'(N, D)| = |\mathbb{B}(N-1, D)| + |\mathbb{B}(N-1, D-1)| = \binom{N}{D}.$$

In particular, $\mathbb{B}'(2D, D)$ contains the *middle levels*. Strings in $\mathbb{B}'(N, D)$ and $\mathbb{B}(N, D)$ are *shorthand* and *longhand representations* of one another.

A *fixed-density de Bruijn cycle* is a length $\binom{N}{D}$ string containing every string in $\mathbb{B}'(N, D)$ exactly once as a (circular) substring. For example,

$$00011100110101001011$$

is a fixed-density de Bruijn cycle for $N = 6$ and $D = 3$ from its substrings

$$000111, 001110, 011100, 111000, 110010, 100110, 001101, 011010, 110100, 101010,$$
$$010101, 101001, 010011, 100101, 001011, 010110, 101101, 011001, 110001, 100011$$

Successive longhand strings differ by length $N-1$ or N *prefix right-shifts*. That is, if $\mathbf{t}$ follows $\mathbf{s}$ then $\mathbf{t} = \overrightarrow{\text{shift}}(\mathbf{s}, 1, j)$ for $j \in \{N-1, N\}$.

Question

Do fixed-density de Bruijn cycles exist for all N and D?

Fixed-Density de Bruijn Cycles

Let $\mathbb{B}'(N, D) = \mathbb{B}(N-1, D) \cup \mathbb{B}(N-1, D-1)$.

$$|\mathbb{B}'(N, D)| = |\mathbb{B}(N-1, D)| + |\mathbb{B}(N-1, D-1)| = \binom{N}{D}.$$

In particular, $\mathbb{B}'(2D, D)$ contains the *middle levels*. Strings in $\mathbb{B}'(N, D)$ and $\mathbb{B}(N, D)$ are *shorthand* and *longhand representations* of one another.

A *fixed-density de Bruijn cycle* is a length $\binom{N}{D}$ string containing every string in $\mathbb{B}'(N, D)$ exactly once as a (circular) substring. For example,

$$00011100110101001011$$

is a fixed-density de Bruijn cycle for $N = 6$ and $D = 3$ from its substrings

$$000111, 001110, 011100, 111000, 110010, 100110, 001101, 011010, 110100, 101010,$$
$$010101, 101001, 010011, 100101, 001011, 010110, 101101, 011001, 110001, 100011$$

Successive longhand strings differ by length $N-1$ or N *prefix right-shifts*. That is, if $\mathbf{t}$ follows $\mathbf{s}$ then $\mathbf{t} = \overrightarrow{\text{shift}}(\mathbf{s}, 1, j)$ for $j \in \{N-1, N\}$.

Question

Do fixed-density de Bruijn cycles exist for all N and D?

Fixed-Density de Bruijn Cycles

Let $\mathbb{B}'(N, D) = \mathbb{B}(N-1, D) \cup \mathbb{B}(N-1, D-1)$.

$$|\mathbb{B}'(N, D)| = |\mathbb{B}(N-1, D)| + |\mathbb{B}(N-1, D-1)| = \binom{N}{D}.$$

In particular, $\mathbb{B}'(2D, D)$ contains the *middle levels*. Strings in $\mathbb{B}'(N, D)$ and $\mathbb{B}(N, D)$ are *shorthand* and *longhand representations* of one another.

A *fixed-density de Bruijn cycle* is a length $\binom{N}{D}$ string containing every string in $\mathbb{B}'(N, D)$ exactly once as a (circular) substring. For example,

$$0001110011010101001011$$

is a fixed-density de Bruijn cycle for $N = 6$ and $D = 3$ from its substrings

$$000111, 001110, 011100, 111000, 110010, 100110, 001101, 011010, 110100, 101010,$$
$$010101, 101001, 010011, 100101, 001011, 010110, 101101, 011001, 110001, 100011$$

Successive longhand strings differ by length $N-1$ or N *prefix right-shifts*. That is, if $\mathbf{t}$ follows $\mathbf{s}$ then $\mathbf{t} = \overrightarrow{\text{shift}}(\mathbf{s}, 1, j)$ for $j \in \{N-1, N\}$.

Question

Do fixed-density de Bruijn cycles exist for all N and D?

Fixed-Density de Bruijn Cycles

Let $\mathbb{B}'(N, D) = \mathbb{B}(N-1, D) \cup \mathbb{B}(N-1, D-1)$.

$$|\mathbb{B}'(N, D)| = |\mathbb{B}(N-1, D)| + |\mathbb{B}(N-1, D-1)| = \binom{N}{D}.$$

In particular, $\mathbb{B}'(2D, D)$ contains the *middle levels*. Strings in $\mathbb{B}'(N, D)$ and $\mathbb{B}(N, D)$ are *shorthand* and *longhand representations* of one another.

A *fixed-density de Bruijn cycle* is a length $\binom{N}{D}$ string containing every string in $\mathbb{B}'(N, D)$ exactly once as a (circular) substring. For example,

$$00011100110101001011$$

is a fixed-density de Bruijn cycle for $N = 6$ and $D = 3$ from its substrings

$$000111, 001110, 011100, 111000, 110010, 100110, 001101, 011010, 110100, 101010,$$
$$010101, 101001, 010011, 100101, 001011, 010110, 101101, 011001, 110001, 100011$$

Successive longhand strings differ by length $N-1$ or N *prefix right-shifts*. That is, if $\mathbf{t}$ follows $\mathbf{s}$ then $\mathbf{t} = \overrightarrow{\text{shift}}(\mathbf{s}, 1, j)$ for $j \in \{N-1, N\}$.

Question

Do fixed-density de Bruijn cycles exist for all N and D?

Fixed-Density de Bruijn Cycles

Let $\mathbb{B}'(N, D) = \mathbb{B}(N-1, D) \cup \mathbb{B}(N-1, D-1)$.

$$|\mathbb{B}'(N, D)| = |\mathbb{B}(N-1, D)| + |\mathbb{B}(N-1, D-1)| = \binom{N}{D}.$$

In particular, $\mathbb{B}'(2D, D)$ contains the *middle levels*. Strings in $\mathbb{B}'(N, D)$ and $\mathbb{B}(N, D)$ are *shorthand* and *longhand representations* of one another.

A *fixed-density de Bruijn cycle* is a length $\binom{N}{D}$ string containing every string in $\mathbb{B}'(N, D)$ exactly once as a (circular) substring. For example,

$$000111001101010011011$$

is a fixed-density de Bruijn cycle for $N = 6$ and $D = 3$ from its substrings

$$000111, 001110, 011100, 111000, 110010, 100110, 001101, 011010, 110100, 101010,$$
$$010101, 101001, 010011, 100101, 001011, 010110, 101101, 011001, 110001, 100011$$

Successive longhand strings differ by length $N-1$ or N *prefix right-shifts*. That is, if $\mathbf{t}$ follows $\mathbf{s}$ then $\mathbf{t} = \overrightarrow{\text{shift}}(\mathbf{s}, 1, j)$ for $j \in \{N-1, N\}$.

Question

Do fixed-density de Bruijn cycles exist for all N and D?

Fixed-Density de Bruijn Cycles

Let $\mathbb{B}'(N, D) = \mathbb{B}(N-1, D) \cup \mathbb{B}(N-1, D-1)$.

$$|\mathbb{B}'(N, D)| = |\mathbb{B}(N-1, D)| + |\mathbb{B}(N-1, D-1)| = \binom{N}{D}.$$

In particular, $\mathbb{B}'(2D, D)$ contains the *middle levels*. Strings in $\mathbb{B}'(N, D)$ and $\mathbb{B}(N, D)$ are *shorthand* and *longhand representations* of one another.

A *fixed-density de Bruijn cycle* is a length $\binom{N}{D}$ string containing every string in $\mathbb{B}'(N, D)$ exactly once as a (circular) substring. For example,

$$00011100110101001011$$

is a fixed-density de Bruijn cycle for $N = 6$ and $D = 3$ from its substrings

$$000111, 001110, 011100, 111000, 110010, 100110, 001101, 011010, 110100, 101010,$$
$$010101, 101001, 010011, 100101, 001011, 010110, 101101, 011001, 110001, 100011$$

Successive longhand strings differ by length $N-1$ or N *prefix right-shifts*. That is, if $\mathbf{t}$ follows $\mathbf{s}$ then $\mathbf{t} = \overrightarrow{\text{shift}}(\mathbf{s}, 1, j)$ for $j \in \{N-1, N\}$.

Question

Do fixed-density de Bruijn cycles exist for all N and D?

Fixed-Density de Bruijn Cycles

Let $\mathbb{B}'(N, D) = \mathbb{B}(N-1, D) \cup \mathbb{B}(N-1, D-1)$.

$$|\mathbb{B}'(N, D)| = |\mathbb{B}(N-1, D)| + |\mathbb{B}(N-1, D-1)| = \binom{N}{D}.$$

In particular, $\mathbb{B}'(2D, D)$ contains the *middle levels*. Strings in $\mathbb{B}'(N, D)$ and $\mathbb{B}(N, D)$ are *shorthand* and *longhand representations* of one another.

A *fixed-density de Bruijn cycle* is a length $\binom{N}{D}$ string containing every string in $\mathbb{B}'(N, D)$ exactly once as a (circular) substring. For example,

$$00011100110101001011$$

is a fixed-density de Bruijn cycle for $N = 6$ and $D = 3$ from its substrings

$$000111, 001110, 011100, 111000, 110010, 100110, 001101, 011010, 110100, 101010,$$
$$010101, 101001, 010011, 100101, 001011, 010110, 101101, 011001, 110001, 100011$$

Successive longhand strings differ by length $N-1$ or N *prefix right-shifts*. That is, if $\mathbf{t}$ follows $\mathbf{s}$ then $\mathbf{t} = \overrightarrow{\text{shift}}(\mathbf{s}, 1, j)$ for $j \in \{N-1, N\}$.

Question

Do fixed-density de Bruijn cycles exist for all N and D?

Fixed-Density de Bruijn Cycles

Let $\mathbb{B}'(N, D) = \mathbb{B}(N-1, D) \cup \mathbb{B}(N-1, D-1)$.

$$|\mathbb{B}'(N, D)| = |\mathbb{B}(N-1, D)| + |\mathbb{B}(N-1, D-1)| = \binom{N}{D}.$$

In particular, $\mathbb{B}'(2D, D)$ contains the *middle levels*. Strings in $\mathbb{B}'(N, D)$ and $\mathbb{B}(N, D)$ are *shorthand* and *longhand representations* of one another.

A *fixed-density de Bruijn cycle* is a length $\binom{N}{D}$ string containing every string in $\mathbb{B}'(N, D)$ exactly once as a (circular) substring. For example,

$$000111001101010010 11$$

is a fixed-density de Bruijn cycle for $N = 6$ and $D = 3$ from its substrings

$000111, 001110, 011100, 111000, 110010, 100110, 001101, 011010, 110100, 101010,$
$010101, 101001, 010011, 100101, 001011, 010110, 101101, 011001, 110001, 100011$

Successive longhand strings differ by length $N-1$ or N *prefix right-shifts*. That is, if $\mathbf{t}$ follows $\mathbf{s}$ then $\mathbf{t} = \overrightarrow{\text{shift}}(\mathbf{s}, 1, j)$ for $j \in \{N-1, N\}$.

Question

Do fixed-density de Bruijn cycles exist for all N and D?

Fixed-Density de Bruijn Cycles

Let $\mathbb{B}'(N, D) = \mathbb{B}(N-1, D) \cup \mathbb{B}(N-1, D-1)$.

$$|\mathbb{B}'(N, D)| = |\mathbb{B}(N-1, D)| + |\mathbb{B}(N-1, D-1)| = \binom{N}{D}.$$

In particular, $\mathbb{B}'(2D, D)$ contains the *middle levels*. Strings in $\mathbb{B}'(N, D)$ and $\mathbb{B}(N, D)$ are *shorthand* and *longhand representations* of one another.

A *fixed-density de Bruijn cycle* is a length $\binom{N}{D}$ string containing every string in $\mathbb{B}'(N, D)$ exactly once as a (circular) substring. For example,

$$0001110011010100 1011$$

is a fixed-density de Bruijn cycle for $N = 6$ and $D = 3$ from its substrings

$000111, 001110, 011100, 111000, 110010, 100110, 001101, 011010, 110100, 101010,$
$010101, 101001, 010011, 100101, 001011, 010110, 101101, 011001, 110001, 100011$

Successive longhand strings differ by length $N-1$ or N *prefix right-shifts*. That is, if $\mathbf{t}$ follows $\mathbf{s}$ then $\mathbf{t} = \overrightarrow{\text{shift}}(\mathbf{s}, 1, j)$ for $j \in \{N-1, N\}$.

Question

Do fixed-density de Bruijn cycles exist for all N and D?

Fixed-Density de Bruijn Cycles

Let $\mathbb{B}'(N, D) = \mathbb{B}(N-1, D) \cup \mathbb{B}(N-1, D-1)$.

$$|\mathbb{B}'(N, D)| = |\mathbb{B}(N-1, D)| + |\mathbb{B}(N-1, D-1)| = \binom{N}{D}.$$

In particular, $\mathbb{B}'(2D, D)$ contains the *middle levels*. Strings in $\mathbb{B}'(N, D)$ and $\mathbb{B}(N, D)$ are *shorthand* and *longhand representations* of one another.

A *fixed-density de Bruijn cycle* is a length $\binom{N}{D}$ string containing every string in $\mathbb{B}'(N, D)$ exactly once as a (circular) substring. For example,

$$00011100110101001011$$

is a fixed-density de Bruijn cycle for $N = 6$ and $D = 3$ from its substrings

$000111, 001110, 011100, 111000, 110010, 100110, 001101, 011010, 110100, 101010,$
$010101, 101001, 010011, 100101, 001011, 010110, 101101, 011001, 110001, 100011$

Successive longhand strings differ by length $N-1$ or N *prefix right-shifts*. That is, if $\mathbf{t}$ follows $\mathbf{s}$ then $\mathbf{t} = \overrightarrow{\text{shift}}(\mathbf{s}, 1, j)$ for $j \in \{N-1, N\}$.

Question

Do fixed-density de Bruijn cycles exist for all N and D?

Fixed-Density de Bruijn Cycles

Let $\mathbb{B}'(N, D) = \mathbb{B}(N-1, D) \cup \mathbb{B}(N-1, D-1)$.

$$|\mathbb{B}'(N, D)| = |\mathbb{B}(N-1, D)| + |\mathbb{B}(N-1, D-1)| = \binom{N}{D}.$$

In particular, $\mathbb{B}'(2D, D)$ contains the *middle levels*. Strings in $\mathbb{B}'(N, D)$ and $\mathbb{B}(N, D)$ are *shorthand* and *longhand representations* of one another.

A *fixed-density de Bruijn cycle* is a length $\binom{N}{D}$ string containing every string in $\mathbb{B}'(N, D)$ exactly once as a (circular) substring. For example,

$$000111100110101001011$$

is a fixed-density de Bruijn cycle for $N = 6$ and $D = 3$ from its substrings

$$000111, 001110, 011100, 111000, 110010, 100110, 001101, 011010, 110100, 101010,$$
$$010101, 101001, 010011, 100101, 001011, 010110, 101101, 011001, 110001, 100011$$

Successive longhand strings differ by length $N-1$ or N *prefix right-shifts*. That is, if $\mathbf{t}$ follows $\mathbf{s}$ then $\mathbf{t} = \overrightarrow{\text{shift}}(\mathbf{s}, 1, j)$ for $j \in \{N-1, N\}$.

Question

Do fixed-density de Bruijn cycles exist for all N and D?

Fixed-Density de Bruijn Cycles

Let $\mathbb{B}'(N, D) = \mathbb{B}(N-1, D) \cup \mathbb{B}(N-1, D-1)$.

$$|\mathbb{B}'(N, D)| = |\mathbb{B}(N-1, D)| + |\mathbb{B}(N-1, D-1)| = \binom{N}{D}.$$

In particular, $\mathbb{B}'(2D, D)$ contains the *middle levels*. Strings in $\mathbb{B}'(N, D)$ and $\mathbb{B}(N, D)$ are *shorthand* and *longhand representations* of one another.

A *fixed-density de Bruijn cycle* is a length $\binom{N}{D}$ string containing every string in $\mathbb{B}'(N, D)$ exactly once as a (circular) substring. For example,

$$0001110011010101001011$$

is a fixed-density de Bruijn cycle for $N = 6$ and $D = 3$ from its substrings

$000111, 001110, 011100, 111000, 110010, 100110, 001101, 011010, 110100, 101010,$
$010101, 101001, 010011, 100101, 001011, 010110, 101101, 011001, 110001, 100011$

Successive longhand strings differ by length $N-1$ or N *prefix right-shifts*. That is, if $\mathbf{t}$ follows $\mathbf{s}$ then $\mathbf{t} = \overrightarrow{\text{shift}}(\mathbf{s}, 1, j)$ for $j \in \{N-1, N\}$.

Question

Do fixed-density de Bruijn cycles exist for all N and D?

Fixed-Density de Bruijn Cycles

Let $\mathbb{B}'(N, D) = \mathbb{B}(N-1, D) \cup \mathbb{B}(N-1, D-1)$.

$$|\mathbb{B}'(N, D)| = |\mathbb{B}(N-1, D)| + |\mathbb{B}(N-1, D-1)| = \binom{N}{D}.$$

In particular, $\mathbb{B}'(2D, D)$ contains the *middle levels*. Strings in $\mathbb{B}'(N, D)$ and $\mathbb{B}(N, D)$ are *shorthand* and *longhand representations* of one another.

A *fixed-density de Bruijn cycle* is a length $\binom{N}{D}$ string containing every string in $\mathbb{B}'(N, D)$ exactly once as a (circular) substring. For example,

$$00011100110101001011$$

is a fixed-density de Bruijn cycle for $N = 6$ and $D = 3$ from its substrings

$$000111, 001110, 011100, 111000, 110010, 100110, 001101, 011010, 110100, 101010,$$
$$010101, 101001, 010011, 100101, 001011, 010110, 101101, 011001, 110001, 100011$$

Successive longhand strings differ by length $N-1$ or N *prefix right-shifts*. That is, if $\mathbf{t}$ follows $\mathbf{s}$ then $\mathbf{t} = \overrightarrow{\text{shift}}(\mathbf{s}, 1, j)$ for $j \in \{N-1, N\}$.

Question

Do fixed-density de Bruijn cycles exist for all N and D?

Fixed-Density de Bruijn Cycles

Let $\mathbb{B}'(N, D) = \mathbb{B}(N-1, D) \cup \mathbb{B}(N-1, D-1)$.

$$|\mathbb{B}'(N, D)| = |\mathbb{B}(N-1, D)| + |\mathbb{B}(N-1, D-1)| = \binom{N}{D}.$$

In particular, $\mathbb{B}'(2D, D)$ contains the *middle levels*. Strings in $\mathbb{B}'(N, D)$ and $\mathbb{B}(N, D)$ are *shorthand* and *longhand representations* of one another.

A *fixed-density de Bruijn cycle* is a length $\binom{N}{D}$ string containing every string in $\mathbb{B}'(N, D)$ exactly once as a (circular) substring. For example,

$$00011100110101001011$$

is a fixed-density de Bruijn cycle for $N = 6$ and $D = 3$ from its substrings

$$000111, 001110, 011100, 111000, 110010, 100110, 001101, 011010, 110100, 101010,$$
$$010101, 101001, 010011, 100101, 001011, 010110, 101101, 011001, 110001, 100011$$

Successive longhand strings differ by length $N-1$ or N *prefix right-shifts*. That is, if $\mathbf{t}$ follows $\mathbf{s}$ then $\mathbf{t} = \overrightarrow{\text{shift}}(\mathbf{s}, 1, j)$ for $j \in \{N-1, N\}$.

Question

Do fixed-density de Bruijn cycles exist for all N and D?

Fixed-Density de Bruijn Cycles

Let $\mathbb{B}'(N, D) = \mathbb{B}(N-1, D) \cup \mathbb{B}(N-1, D-1)$.

$$|\mathbb{B}'(N, D)| = |\mathbb{B}(N-1, D)| + |\mathbb{B}(N-1, D-1)| = \binom{N}{D}.$$

In particular, $\mathbb{B}'(2D, D)$ contains the *middle levels*. Strings in $\mathbb{B}'(N, D)$ and $\mathbb{B}(N, D)$ are *shorthand* and *longhand representations* of one another.

A *fixed-density de Bruijn cycle* is a length $\binom{N}{D}$ string containing every string in $\mathbb{B}'(N, D)$ exactly once as a (circular) substring. For example,

$$00011100110101001011$$

is a fixed-density de Bruijn cycle for $N = 6$ and $D = 3$ from its substrings

$$000111, 001110, 011100, 111000, 110010, 100110, 001101, 011010, 110100, 101010,$$
$$010101, 101001, 010011, 100101, 001011, 010110, 101101, 011001, 110001, 100011$$

Successive longhand strings differ by length $N-1$ or N *prefix right-shifts*. That is, if $\mathbf{t}$ follows $\mathbf{s}$ then $\mathbf{t} = \overrightarrow{\text{shift}}(\mathbf{s}, 1, j)$ for $j \in \{N-1, N\}$.

Question

Do fixed-density de Bruijn cycles exist for all N and D?

Fixed-Density de Bruijn Cycles

Let $\mathbb{B}'(N, D) = \mathbb{B}(N-1, D) \cup \mathbb{B}(N-1, D-1)$.

$$|\mathbb{B}'(N, D)| = |\mathbb{B}(N-1, D)| + |\mathbb{B}(N-1, D-1)| = \binom{N}{D}.$$

In particular, $\mathbb{B}'(2D, D)$ contains the *middle levels*. Strings in $\mathbb{B}'(N, D)$ and $\mathbb{B}(N, D)$ are *shorthand* and *longhand representations* of one another.

A *fixed-density de Bruijn cycle* is a length $\binom{N}{D}$ string containing every string in $\mathbb{B}'(N, D)$ exactly once as a (circular) substring. For example,

$$00011100110101001011$$

is a fixed-density de Bruijn cycle for $N = 6$ and $D = 3$ from its substrings

$$000111, 001110, 011100, 111000, 110010, 100110, 001101, 011010, 110100, 101010,$$
$$010101, 101001, 010011, 100101, 001011, 010110, 101101, 011001, 110001, 100011$$

Successive longhand strings differ by length $N-1$ or N *prefix right-shifts*. That is, if $\mathbf{t}$ follows $\mathbf{s}$ then $\mathbf{t} = \overrightarrow{\text{shift}}(\mathbf{s}, 1, j)$ for $j \in \{N-1, N\}$.

Question

Do fixed-density de Bruijn cycles exist for all N and D?

Fixed-Density de Bruijn Cycles

Let $\mathbb{B}'(N, D) = \mathbb{B}(N-1, D) \cup \mathbb{B}(N-1, D-1)$.

$$|\mathbb{B}'(N, D)| = |\mathbb{B}(N-1, D)| + |\mathbb{B}(N-1, D-1)| = \binom{N}{D}.$$

In particular, $\mathbb{B}'(2D, D)$ contains the *middle levels*. Strings in $\mathbb{B}'(N, D)$ and $\mathbb{B}(N, D)$ are *shorthand* and *longhand representations* of one another.

A *fixed-density de Bruijn cycle* is a length $\binom{N}{D}$ string containing every string in $\mathbb{B}'(N, D)$ exactly once as a (circular) substring. For example,

$$00011100110101001011$$

is a fixed-density de Bruijn cycle for $N = 6$ and $D = 3$ from its substrings

$$000111, 001110, 011100, 111000, 110010, 100110, 001101, 011010, 110100, 101010,$$
$$010101, 101001, 010011, 100101, 001011, 010110, 101101, 011001, 110001, 100011$$

Successive longhand strings differ by length $N-1$ or N *prefix right-shifts*. That is, if $\mathbf{t}$ follows $\mathbf{s}$ then $\mathbf{t} = \overrightarrow{\text{shift}}(\mathbf{s}, 1, j)$ for $j \in \{N-1, N\}$.

Question

Do fixed-density de Bruijn cycles exist for all N and D?

Fixed-Density de Bruijn Cycles

Let $\mathbb{B}'(N, D) = \mathbb{B}(N-1, D) \cup \mathbb{B}(N-1, D-1)$.

$$|\mathbb{B}'(N, D)| = |\mathbb{B}(N-1, D)| + |\mathbb{B}(N-1, D-1)| = \binom{N}{D}.$$

In particular, $\mathbb{B}'(2D, D)$ contains the *middle levels*. Strings in $\mathbb{B}'(N, D)$ and $\mathbb{B}(N, D)$ are *shorthand* and *longhand representations* of one another.

A *fixed-density de Bruijn cycle* is a length $\binom{N}{D}$ string containing every string in $\mathbb{B}'(N, D)$ exactly once as a (circular) substring. For example,

$$00011100110101001011$$

is a fixed-density de Bruijn cycle for $N = 6$ and $D = 3$ from its substrings

$$000111, 001110, 011100, 111000, 110010, 100110, 001101, 011010, 110100, 101010,$$
$$010101, 101001, 010011, 100101, 001011, 010110, 101101, 011001, 110001, 100011$$

Successive longhand strings differ by length $N-1$ or N *prefix right-shifts*. That is, if $\mathbf{t}$ follows $\mathbf{s}$ then $\mathbf{t} = \overrightarrow{\text{shift}}(\mathbf{s}, 1, j)$ for $j \in \{N-1, N\}$.

Question

Do fixed-density de Bruijn cycles exist for all N and D?

Fixed-Density de Bruijn Cycles

Let $\mathbb{B}'(N, D) = \mathbb{B}(N-1, D) \cup \mathbb{B}(N-1, D-1)$.

$$|\mathbb{B}'(N, D)| = |\mathbb{B}(N-1, D)| + |\mathbb{B}(N-1, D-1)| = \binom{N}{D}.$$

In particular, $\mathbb{B}'(2D, D)$ contains the *middle levels*. Strings in $\mathbb{B}'(N, D)$ and $\mathbb{B}(N, D)$ are *shorthand* and *longhand representations* of one another.

A *fixed-density de Bruijn cycle* is a length $\binom{N}{D}$ string containing every string in $\mathbb{B}'(N, D)$ exactly once as a (circular) substring. For example,

$$0001110011010100 1011$$

is a fixed-density de Bruijn cycle for $N = 6$ and $D = 3$ from its substrings

$000111, 001110, 011100, 111000, 110010, 100110, 001101, 011010, 110100, 101010,$
$010101, 101001, 010011, 100101, 001011, 010110, 101101, 011001, 110001, 100011$

Successive longhand strings differ by length $N-1$ or N *prefix right-shifts*. That is, if $\mathbf{t}$ follows $\mathbf{s}$ then $\mathbf{t} = \overrightarrow{\text{shift}}(\mathbf{s}, 1, j)$ for $j \in \{N-1, N\}$.

Question

Do fixed-density de Bruijn cycles exist for all N and D?

Fixed-Density de Bruijn Cycles

Let $\mathbb{B}'(N, D) = \mathbb{B}(N-1, D) \cup \mathbb{B}(N-1, D-1)$.

$$|\mathbb{B}'(N, D)| = |\mathbb{B}(N-1, D)| + |\mathbb{B}(N-1, D-1)| = \binom{N}{D}.$$

In particular, $\mathbb{B}'(2D, D)$ contains the *middle levels*. Strings in $\mathbb{B}'(N, D)$ and $\mathbb{B}(N, D)$ are *shorthand* and *longhand representations* of one another.

A *fixed-density de Bruijn cycle* is a length $\binom{N}{D}$ string containing every string in $\mathbb{B}'(N, D)$ exactly once as a (circular) substring. For example,

$$00011100110101001011$$

is a fixed-density de Bruijn cycle for $N = 6$ and $D = 3$ from its substrings

$$000111, 001110, 011100, 111000, 110010, 100110, 001101, 011010, 110100, 101010,$$
$$010101, 101001, 010011, 100101, 001011, 010110, 101101, 011001, 110001, 100011$$

Successive longhand strings differ by length $N-1$ or N *prefix right-shifts*. That is, if $\mathbf{t}$ follows $\mathbf{s}$ then $\mathbf{t} = \overrightarrow{\text{shift}}(\mathbf{s}, 1, j)$ for $j \in \{N-1, N\}$.

Question

Do fixed-density de Bruijn cycles exist for all N and D?

Fixed-Density de Bruijn Cycles

Let $\mathbb{B}'(N, D) = \mathbb{B}(N-1, D) \cup \mathbb{B}(N-1, D-1)$.

$$|\mathbb{B}'(N, D)| = |\mathbb{B}(N-1, D)| + |\mathbb{B}(N-1, D-1)| = \binom{N}{D}.$$

In particular, $\mathbb{B}'(2D, D)$ contains the *middle levels*. Strings in $\mathbb{B}'(N, D)$ and $\mathbb{B}(N, D)$ are *shorthand* and *longhand representations* of one another.

A *fixed-density de Bruijn cycle* is a length $\binom{N}{D}$ string containing every string in $\mathbb{B}'(N, D)$ exactly once as a (circular) substring. For example,

$$00011100110101001011$$

is a fixed-density de Bruijn cycle for $N = 6$ and $D = 3$ from its substrings

$$000111, 001110, 011100, 111000, 110010, 100110, 001101, 011010, 110100, 101010,$$
$$010101, 101001, 010011, 100101, 001011, 010110, 101101, 011001, 110001, 100011$$

Successive longhand strings differ by length $N-1$ or N *prefix right-shifts*. That is, if $\mathbf{t}$ follows $\mathbf{s}$ then $\mathbf{t} = \overrightarrow{\text{shift}}(\mathbf{s}, 1, j)$ for $j \in \{N-1, N\}$.

Question

Do fixed-density de Bruijn cycles exist for all N and D?

Fixed-Density de Bruijn Cycles

Let $\mathbb{B}'(N, D) = \mathbb{B}(N-1, D) \cup \mathbb{B}(N-1, D-1)$.

$$|\mathbb{B}'(N, D)| = |\mathbb{B}(N-1, D)| + |\mathbb{B}(N-1, D-1)| = \binom{N}{D}.$$

In particular, $\mathbb{B}'(2D, D)$ contains the *middle levels*. Strings in $\mathbb{B}'(N, D)$ and $\mathbb{B}(N, D)$ are *shorthand* and *longhand representations* of one another.

A *fixed-density de Bruijn cycle* is a length $\binom{N}{D}$ string containing every string in $\mathbb{B}'(N, D)$ exactly once as a (circular) substring. For example,

$$00011100110101001011$$

is a fixed-density de Bruijn cycle for $N = 6$ and $D = 3$ from its substrings

$$000111, 001110, 011100, 111000, 110010, 100110, 001101, 011010, 110100, 101010,$$
$$010101, 101001, 010011, 100101, 001011, 010110, 101101, 011001, 110001, 100011$$

Successive longhand strings differ by length $N-1$ or N *prefix right-shifts*. That is, if $\mathbf{t}$ follows $\mathbf{s}$ then $\mathbf{t} = \overrightarrow{\text{shift}}(\mathbf{s}, 1, j)$ for $j \in \{N-1, N\}$.

Question

Do fixed-density de Bruijn cycles exist for all N and D?

Fixed-Density de Bruijn Cycles

Let $\mathbb{B}'(N, D) = \mathbb{B}(N-1, D) \cup \mathbb{B}(N-1, D-1)$.

$$|\mathbb{B}'(N, D)| = |\mathbb{B}(N-1, D)| + |\mathbb{B}(N-1, D-1)| = \binom{N}{D}.$$

In particular, $\mathbb{B}'(2D, D)$ contains the *middle levels*. Strings in $\mathbb{B}'(N, D)$ and $\mathbb{B}(N, D)$ are *shorthand* and *longhand representations* of one another.

A *fixed-density de Bruijn cycle* is a length $\binom{N}{D}$ string containing every string in $\mathbb{B}'(N, D)$ exactly once as a (circular) substring. For example,

$$0001110011010101011$$

is a fixed-density de Bruijn cycle for $N = 6$ and $D = 3$ from its substrings

$$000111, 001110, 011100, 111000, 110010, 100110, 001101, 011010, 110100, 101010,$$
$$010101, 101001, 010011, 100101, 001011, 010110, 101101, 011001, 110001, 100011$$

Successive longhand strings differ by length $N-1$ or N *prefix right-shifts*. That is, if $\mathbf{t}$ follows $\mathbf{s}$ then $\mathbf{t} = \overrightarrow{\text{shift}}(\mathbf{s}, 1, j)$ for $j \in \{N-1, N\}$.

Question

Do fixed-density de Bruijn cycles exist for all N and D?

Fixed-Density de Bruijn Cycles

Let $\mathbb{B}'(N, D) = \mathbb{B}(N-1, D) \cup \mathbb{B}(N-1, D-1)$.

$$|\mathbb{B}'(N, D)| = |\mathbb{B}(N-1, D)| + |\mathbb{B}(N-1, D-1)| = \binom{N}{D}.$$

In particular, $\mathbb{B}'(2D, D)$ contains the *middle levels*. Strings in $\mathbb{B}'(N, D)$ and $\mathbb{B}(N, D)$ are *shorthand* and *longhand representations* of one another.

A *fixed-density de Bruijn cycle* is a length $\binom{N}{D}$ string containing every string in $\mathbb{B}'(N, D)$ exactly once as a (circular) substring. For example,

$$0001110011010100 1011$$

is a fixed-density de Bruijn cycle for $N = 6$ and $D = 3$ from its substrings

$$000111, 001110, 011100, 111000, 110010, 100110, 001101, 011010, 110100, 101010,$$
$$010101, 101001, 010011, 100101, 001011, 010110, 101101, 011001, 110001, 100011$$

Successive longhand strings differ by length $N-1$ or N *prefix right-shifts*. That is, if $\mathbf{t}$ follows $\mathbf{s}$ then $\mathbf{t} = \overrightarrow{\text{shift}}(\mathbf{s}, 1, j)$ for $j \in \{N-1, N\}$.

Question

Do fixed-density de Bruijn cycles exist for all N and D?

Fixed-Density de Bruijn Cycles

Let $\mathbb{B}'(N, D) = \mathbb{B}(N-1, D) \cup \mathbb{B}(N-1, D-1)$.

$$|\mathbb{B}'(N, D)| = |\mathbb{B}(N-1, D)| + |\mathbb{B}(N-1, D-1)| = \binom{N}{D}.$$

In particular, $\mathbb{B}'(2D, D)$ contains the *middle levels*. Strings in $\mathbb{B}'(N, D)$ and $\mathbb{B}(N, D)$ are *shorthand* and *longhand representations* of one another.

A *fixed-density de Bruijn cycle* is a length $\binom{N}{D}$ string containing every string in $\mathbb{B}'(N, D)$ exactly once as a (circular) substring. For example,

$$00011100110101001011$$

is a fixed-density de Bruijn cycle for $N = 6$ and $D = 3$ from its substrings

$$000111, 001110, 011100, 111000, 110010, 100110, 001101, 011010, 110100, 101010,$$
$$010101, 101001, 010011, 100101, 001011, 010110, 101101, 011001, 110001, 100011$$

Successive longhand strings differ by length $N-1$ or N *prefix right-shifts*. That is, if $\mathbf{t}$ follows $\mathbf{s}$ then $\mathbf{t} = \overrightarrow{\text{shift}}(\mathbf{s}, 1, j)$ for $j \in \{N-1, N\}$.

Question

Do fixed-density de Bruijn cycles exist for all N and D?

Fixed-Density de Bruijn Cycles

Let $\mathbb{B}'(N, D) = \mathbb{B}(N-1, D) \cup \mathbb{B}(N-1, D-1)$.

$$|\mathbb{B}'(N, D)| = |\mathbb{B}(N-1, D)| + |\mathbb{B}(N-1, D-1)| = \binom{N}{D}.$$

In particular, $\mathbb{B}'(2D, D)$ contains the *middle levels*. Strings in $\mathbb{B}'(N, D)$ and $\mathbb{B}(N, D)$ are *shorthand* and *longhand representations* of one another.

A *fixed-density de Bruijn cycle* is a length $\binom{N}{D}$ string containing every string in $\mathbb{B}'(N, D)$ exactly once as a (circular) substring. For example,

$$000111001101010010 11$$

is a fixed-density de Bruijn cycle for $N = 6$ and $D = 3$ from its substrings

$$000111, 001110, 011100, 111000, 110010, 100110, 001101, 011010, 110100, 101010,$$
$$010101, 101001, 010011, 100101, 001011, 010110, 101101, 011001, 110001, 100011$$

Successive longhand strings differ by length $N-1$ or N *prefix right-shifts*. That is, if $\mathbf{t}$ follows $\mathbf{s}$ then $\mathbf{t} = \overrightarrow{\text{shift}}(\mathbf{s}, 1, j)$ for $j \in \{N-1, N\}$.

Question

Do fixed-density de Bruijn cycles exist for all N and D?

Fixed-Density de Bruijn Cycles

Let $\mathbb{B}'(N, D) = \mathbb{B}(N-1, D) \cup \mathbb{B}(N-1, D-1)$.

$$|\mathbb{B}'(N, D)| = |\mathbb{B}(N-1, D)| + |\mathbb{B}(N-1, D-1)| = \binom{N}{D}.$$

In particular, $\mathbb{B}'(2D, D)$ contains the *middle levels*. Strings in $\mathbb{B}'(N, D)$ and $\mathbb{B}(N, D)$ are *shorthand* and *longhand representations* of one another.

A *fixed-density de Bruijn cycle* is a length $\binom{N}{D}$ string containing every string in $\mathbb{B}'(N, D)$ exactly once as a (circular) substring. For example,

$$0001110011010100101 1$$

is a fixed-density de Bruijn cycle for $N = 6$ and $D = 3$ from its substrings

$$000111, 001110, 011100, 111000, 110010, 100110, 001101, 011010, 110100, 101010,$$
$$010101, 101001, 010011, 100101, 001011, 010110, 101101, 011001, 110001, 100011$$

Successive longhand strings differ by length $N-1$ or N *prefix right-shifts*. That is, if $\mathbf{t}$ follows $\mathbf{s}$ then $\mathbf{t} = \overrightarrow{\text{shift}}(\mathbf{s}, 1, j)$ for $j \in \{N-1, N\}$.

Question

Do fixed-density de Bruijn cycles exist for all N and D?

Fixed-Density de Bruijn Cycles

Let $\mathbb{B}'(N, D) = \mathbb{B}(N-1, D) \cup \mathbb{B}(N-1, D-1)$.

$$|\mathbb{B}'(N, D)| = |\mathbb{B}(N-1, D)| + |\mathbb{B}(N-1, D-1)| = \binom{N}{D}.$$

In particular, $\mathbb{B}'(2D, D)$ contains the *middle levels*. Strings in $\mathbb{B}'(N, D)$ and $\mathbb{B}(N, D)$ are *shorthand* and *longhand representations* of one another.

A *fixed-density de Bruijn cycle* is a length $\binom{N}{D}$ string containing every string in $\mathbb{B}'(N, D)$ exactly once as a (circular) substring. For example,

$$00011100110101001011$$

is a fixed-density de Bruijn cycle for $N = 6$ and $D = 3$ from its substrings

$$000111, 001110, 011100, 111000, 110010, 100110, 001101, 011010, 110100, 101010,$$
$$010101, 101001, 010011, 100101, 001011, 010110, 101101, 011001, 110001, 100011$$

Successive longhand strings differ by length $N-1$ or N *prefix right-shifts*. That is, if $\mathbf{t}$ follows $\mathbf{s}$ then $\mathbf{t} = \overrightarrow{\text{shift}}(\mathbf{s}, 1, j)$ for $j \in \{N-1, N\}$.

Question

Do fixed-density de Bruijn cycles exist for all N and D?

Fixed-Density de Bruijn Cycles

Let $\mathbb{B}'(N, D) = \mathbb{B}(N-1, D) \cup \mathbb{B}(N-1, D-1)$.

$$|\mathbb{B}'(N, D)| = |\mathbb{B}(N-1, D)| + |\mathbb{B}(N-1, D-1)| = \binom{N}{D}.$$

In particular, $\mathbb{B}'(2D, D)$ contains the *middle levels*. Strings in $\mathbb{B}'(N, D)$ and $\mathbb{B}(N, D)$ are *shorthand* and *longhand representations* of one another.

A *fixed-density de Bruijn cycle* is a length $\binom{N}{D}$ string containing every string in $\mathbb{B}'(N, D)$ exactly once as a (circular) substring. For example,

$$00011100110101001011$$

is a fixed-density de Bruijn cycle for $N = 6$ and $D = 3$ from its substrings

$$000111, 001110, 011100, 111000, 110010, 100110, 001101, 011010, 110100, 101010,$$
$$010101, 101001, 010011, 100101, 001011, 010110, 101101, 011001, 110001, 100011$$

Successive longhand strings differ by length $N-1$ or N *prefix right-shifts*. That is, if $\mathbf{t}$ follows $\mathbf{s}$ then $\mathbf{t} = \overrightarrow{\text{shift}}(\mathbf{s}, 1, j)$ for $j \in \{N-1, N\}$.

Question

Do fixed-density de Bruijn cycles exist for all N and D?

Fixed-Density de Bruijn Cycles

Let $\mathbb{B}'(N, D) = \mathbb{B}(N-1, D) \cup \mathbb{B}(N-1, D-1)$.

$$|\mathbb{B}'(N, D)| = |\mathbb{B}(N-1, D)| + |\mathbb{B}(N-1, D-1)| = \binom{N}{D}.$$

In particular, $\mathbb{B}'(2D, D)$ contains the *middle levels*. Strings in $\mathbb{B}'(N, D)$ and $\mathbb{B}(N, D)$ are *shorthand* and *longhand representations* of one another.

A *fixed-density de Bruijn cycle* is a length $\binom{N}{D}$ string containing every string in $\mathbb{B}'(N, D)$ exactly once as a (circular) substring. For example,

$$00011100110101001011$$

is a fixed-density de Bruijn cycle for $N = 6$ and $D = 3$ from its substrings

$$000111, 001110, 011100, 111000, 110010, 100110, 001101, 011010, 110100, 101010,$$
$$010101, 101001, 010011, 100101, 001011, 010110, 101101, 011001, 110001, 100011$$

Successive longhand strings differ by length $N-1$ or N *prefix right-shifts*. That is, if $\mathbf{t}$ follows $\mathbf{s}$ then $\mathbf{t} = \overrightarrow{\text{shift}}(\mathbf{s}, 1, j)$ for $j \in \{N-1, N\}$.

Question

Do fixed-density de Bruijn cycles exist for all N and D?

Fixed-Density Grand-Daddy?

Can fixed-density de Bruijn cycles be constructed using the lexicographic prime-prefix necklace construction for fixed-density strings? For example,

$$\mathcal{L}(\mathbb{N}(8,4)) = 00001111, 00010111, 00011011, 00011101, 00100111,$$
$$00101011, 00101101, 00110011, 00110101, 01010101$$

so concatenating prime-prefixes gives

$$0000111100010111000110110001110100100111001010110010110100110011010101$$

This is not a fixed-density de Bruijn cycle because it contains the substring $0100100 \notin \mathbb{B}'(8,4)$. This substring comes from the following lexicographically consecutive necklaces in $\mathbb{N}(8,4)$

$$\mathbf{s} = 00011101$$
$$\mathbf{t} = 00100111$$

Notice that the fourth 0 in $\mathbf{t}$ appears two positions to the left of the fourth 0 in $\mathbf{s}$. Therefore, a substring in $\mathbf{s} \cdot \mathbf{t}$ contains too many 0s.

Fixed-Density Grand-Daddy?

Can fixed-density de Bruijn cycles be constructed using the lexicographic prime-prefix necklace construction for fixed-density strings? For example,

$$\mathcal{L}(\mathbb{N}(8,4)) = 00001111, 00010111, 00011011, 00011101, 00100111,$$
$$00101011, 00101101, 00110011, 00110101, 01010101$$

so concatenating prime-prefixes gives

$$00001111000101110001101100011101001001110010101100101101001100110101 01$$

This is not a fixed-density de Bruijn cycle because it contains the substring $0100100 \notin \mathbb{B}'(8,4)$. This substring comes from the following lexicographically consecutive necklaces in $\mathbb{N}(8,4)$

$$\mathbf{s} = 00011101$$
$$\mathbf{t} = 00100111$$

Notice that the fourth 0 in $\mathbf{t}$ appears two positions to the left of the fourth 0 in $\mathbf{s}$. Therefore, a substring in $\mathbf{s} \cdot \mathbf{t}$ contains too many 0s.

Fixed-Density Grand-Daddy?

Can fixed-density de Bruijn cycles be constructed using the lexicographic prime-prefix necklace construction for fixed-density strings? For example,

$$\mathcal{L}(\mathbb{N}(8,4)) = 00001111, 00010111, 00011011, 00011101, 00100111,$$
$$00101011, 00101101, 00110011, 00110101, 01010101$$

so concatenating prime-prefixes gives

$$00001111000101110001101100011101001001110010101100101101001100110101010101$$

This is not a fixed-density de Bruijn cycle because it contains the substring $0100100 \notin \mathbb{B}'(8,4)$. This substring comes from the following lexicographically consecutive necklaces in $\mathbb{N}(8,4)$

$$\mathbf{s} = 00011101$$
$$\mathbf{t} = 00100111$$

Notice that the fourth 0 in $\mathbf{t}$ appears two positions to the left of the fourth 0 in $\mathbf{s}$. Therefore, a substring in $\mathbf{s} \cdot \mathbf{t}$ contains too many 0s.

Fixed-Density Grand-Daddy?

Can fixed-density de Bruijn cycles be constructed using the lexicographic prime-prefix necklace construction for fixed-density strings? For example,

$$\mathcal{L}(\mathbb{N}(8,4)) = 00001111, 00010111, 00011011, 00011101, 00100111,$$
$$00101011, 00101101, 00110011, 00110101, 01010101$$

so concatenating prime-prefixes gives

$$00001111000101110001101100011101001001110010101100101101001100110101$$

This is not a fixed-density de Bruijn cycle because it contains the substring $0100100 \notin \mathbb{B}'(8,4)$. This substring comes from the following lexicographically consecutive necklaces in $\mathbb{N}(8,4)$

$$\mathbf{s} = 00011101$$
$$\mathbf{t} = 00100111$$

Notice that the fourth 0 in $\mathbf{t}$ appears two positions to the left of the fourth 0 in $\mathbf{s}$. Therefore, a substring in $\mathbf{s} \cdot \mathbf{t}$ contains too many 0s.

Fixed-Density Grand-Daddy?

Can fixed-density de Bruijn cycles be constructed using the lexicographic prime-prefix necklace construction for fixed-density strings? For example,

$$\mathcal{L}(\mathbb{N}(8,4)) = 00001111, 00010111, 00011011, 00011101, 00100111,$$
$$00101011, 00101101, 00110011, 00110101, 01010101$$

so concatenating prime-prefixes gives

$$00001111000101110001101100011101001001110010101100101101001100110101010101$$

This is not a fixed-density de Bruijn cycle because it contains the substring $0100100 \notin \mathbb{B}'(8,4)$. This substring comes from the following lexicographically consecutive necklaces in $\mathbb{N}(8,4)$

$$\mathbf{s} = 00011101$$
$$\mathbf{t} = 00100111$$

Notice that the fourth 0 in $\mathbf{t}$ appears two positions to the left of the fourth 0 in $\mathbf{s}$. Therefore, a substring in $\mathbf{s} \cdot \mathbf{t}$ contains too many 0s.

Fixed-Density Grand-Daddy?

Can fixed-density de Bruijn cycles be constructed using the lexicographic prime-prefix necklace construction for fixed-density strings? For example,

$$\mathcal{L}(\mathbb{N}(8,4)) = 00001111, 00010111, 00011011, 00011101, 00100111,$$
$$00101011, 00101101, 00110011, 00110101, 01010101$$

so concatenating prime-prefixes gives

$$00001111000101110001101100011101001001110010101100101101001100110101010101$$

This is not a fixed-density de Bruijn cycle because it contains the substring $0100100 \notin \mathbb{B}'(8,4)$. This substring comes from the following lexicographically consecutive necklaces in $\mathbb{N}(8,4)$

$$\mathbf{s} = 00011101$$
$$\mathbf{t} = 00100111$$

Notice that the fourth 0 in $\mathbf{t}$ appears two positions to the left of the fourth 0 in $\mathbf{s}$. Therefore, a substring in $\mathbf{s} \cdot \mathbf{t}$ contains too many 0s.

Fixed-Density Grand-Daddy?

Can fixed-density de Bruijn cycles be constructed using the lexicographic prime-prefix necklace construction for fixed-density strings? For example,

$$\mathcal{L}(\mathbb{N}(8,4)) = 00001111, 00010111, 00011011, 00011101, 00100111,$$
$$00101011, 00101101, 00110011, 00110101, 01010101$$

so concatenating prime-prefixes gives

$$00001111000101110001101100011101001001110010101100101101001100110101010101$$

This is not a fixed-density de Bruijn cycle because it contains the substring $0100100 \notin \mathbb{B}'(8,4)$. This substring comes from the following lexicographically consecutive necklaces in $\mathbb{N}(8,4)$

$$\mathbf{s} = 00011101$$
$$\mathbf{t} = 00100111$$

Notice that the fourth 0 in $\mathbf{t}$ appears two positions to the left of the fourth 0 in $\mathbf{s}$. Therefore, a substring in $\mathbf{s} \cdot \mathbf{t}$ contains too many 0s.

Fixed-Density Grand-Daddy?

Can fixed-density de Bruijn cycles be constructed using the lexicographic prime-prefix necklace construction for fixed-density strings? For example,

$$\mathcal{L}(\mathbb{N}(8,4)) = 00001111, 00010111, 00011011, 00011101, 00100111,$$
$$00101011, 00101101, 00110011, 00110101, 01010101$$

so concatenating prime-prefixes gives

$$00001111000101110001101100011101001001110010101100101101001100110101010101$$

This is not a fixed-density de Bruijn cycle because it contains the substring $0100100 \notin \mathbb{B}'(8,4)$. This substring comes from the following lexicographically consecutive necklaces in $\mathbb{N}(8,4)$

$$\mathbf{s} = 00011101$$
$$\mathbf{t} = 00100111$$

Notice that the fourth 0 in $\mathbf{t}$ appears two positions to the left of the fourth 0 in $\mathbf{s}$. Therefore, a substring in $\mathbf{s} \cdot \mathbf{t}$ contains too many 0s.

Fixed-Density Grand-Daddy?

Can fixed-density de Bruijn cycles be constructed using the lexicographic prime-prefix necklace construction for fixed-density strings? For example,

$$\mathcal{L}(\mathbb{N}(8,4)) = 00001111, 00010111, 00011011, 00011101, 00100111,$$
$$00101011, 00101101, 00110011, 00110101, 01010101$$

so concatenating prime-prefixes gives

$$00001111000101110001101100011101001001110010101100101101001100110101010101$$

This is not a fixed-density de Bruijn cycle because it contains the substring $0100100 \notin \mathbb{B}'(8,4)$. This substring comes from the following lexicographically consecutive necklaces in $\mathbb{N}(8,4)$

$$\mathbf{s} = 00011101$$
$$\mathbf{t} = 00100111$$

Notice that the fourth 0 in $\mathbf{t}$ appears two positions to the left of the fourth 0 in $\mathbf{s}$. Therefore, a substring in $\mathbf{s} \cdot \mathbf{t}$ contains too many 0s.

Fixed-Density Grand-Daddy?

Can fixed-density de Bruijn cycles be constructed using the lexicographic prime-prefix necklace construction for fixed-density strings? For example,

$$\mathcal{L}(\mathbb{N}(8,4)) = 00001111, 00010111, 00011011, 00011101, 00100111,$$
$$00101011, 00101101, 00110011, 00110101, 01010101$$

so concatenating prime-prefixes gives

$$00001111000101110001101100011101001001110010101100101101001100110101010101$$

This is not a fixed-density de Bruijn cycle because it contains the substring $0100100 \notin \mathbb{B}'(8,4)$. This substring comes from the following lexicographically consecutive necklaces in $\mathbb{N}(8,4)$

$$\mathbf{s} = 00011101$$
$$\mathbf{t} = 00100111$$

Notice that the fourth 0 in $\mathbf{t}$ appears two positions to the left of the fourth 0 in $\mathbf{s}$. Therefore, a substring in $\mathbf{s} \cdot \mathbf{t}$ contains too many 0s.

Fixed-Density Grand-Daddy?

Can fixed-density de Bruijn cycles be constructed using the lexicographic prime-prefix necklace construction for fixed-density strings? For example,

$$\mathcal{L}(\mathbb{N}(8,4)) = 00001111, 00010111, 00011011, 00011101, 00100111,$$
$$00101011, 00101101, 00110011, 00110101, 01010101$$

so concatenating prime-prefixes gives

$$00001111000101110001101100011101001001110010101100101101001100110101010101$$

This is not a fixed-density de Bruijn cycle because it contains the substring $0100100 \notin \mathbb{B}'(8,4)$. This substring comes from the following lexicographically consecutive necklaces in $\mathbb{N}(8,4)$

$$\mathbf{s} = 00011101$$
$$\mathbf{t} = 00100111$$

Notice that the fourth 0 in $\mathbf{t}$ appears two positions to the left of the fourth 0 in $\mathbf{s}$. Therefore, a substring in $\mathbf{s} \cdot \mathbf{t}$ contains too many 0s.

Fixed-Density Grand-Daddy?

Can fixed-density de Bruijn cycles be constructed using the lexicographic prime-prefix necklace construction for fixed-density strings? For example,

$$\mathcal{L}(\mathbb{N}(8,4)) = 00001111, 00010111, 00011011, 00011101, 00100111,$$
$$00101011, 00101101, 00110011, 00110101, 01010101$$

so concatenating prime-prefixes gives

$$00001111000101110001101100011101001001110010101100101101001100110101010101$$

This is not a fixed-density de Bruijn cycle because it contains the substring $0100100 \notin \mathbb{B}'(8,4)$. This substring comes from the following lexicographically consecutive necklaces in $\mathbb{N}(8,4)$

$$\mathbf{s} = 00011101$$
$$\mathbf{t} = 00100111$$

Notice that the fourth 0 in $\mathbf{t}$ appears two positions to the left of the fourth 0 in $\mathbf{s}$. Therefore, a substring in $\mathbf{s} \cdot \mathbf{t}$ contains too many 0s.

Necessary Condition using Right-Shifts

If $\mathbf{s}, \mathbf{t} \in \mathbb{B}(N)$ then $\mathbf{s} \cdot \mathbf{t}$ can appear as a substring of a fixed-density de Bruijn cycle for $\mathbb{N}(N, D)$ only if for all $1 \leq i \leq D$ and $1 \leq j \leq N - D$

$$\text{index}_1(\mathbf{t}, i) \geq \text{index}_1(\mathbf{s}, i) - 1 \text{ and } \text{index}_0(\mathbf{t}, j) \geq \text{index}_0(\mathbf{s}, j) - 1$$

That is, the ith copy of $x \in \{0, 1\}$ can only move to the left at most one position between $\mathbf{s}$ and $\mathbf{t}$. This condition is satisfied iff $\mathbf{s}$ can be transformed into $\mathbf{t}$ by non-overlapping right-shifts.

Unfortunately, the condition is not sufficient. For example, the following is a *(circular) right-shift Gray code* for $\mathbb{N}(7, 3)$

$$0000\overrightarrow{111}, 000\overrightarrow{1}101, 0010\overrightarrow{10}1, 00\overrightarrow{10}011, 000\overrightarrow{1}011$$

Therefore, the (prime-prefix) concatenation gives

$$0000111000110100101010010011000 1011$$

This is not a fixed-density de Bruijn cycle since $010010 \in \mathbb{B}'(7, 3)$ appears twice. Therefore, a special right-shift Gray code will be required.

Necessary Condition using Right-Shifts

If $\mathbf{s}, \mathbf{t} \in \mathbb{B}(N)$ then $\mathbf{s} \cdot \mathbf{t}$ can appear as a substring of a fixed-density de Bruijn cycle for $\mathbb{N}(N, D)$ only if for all $1 \leq i \leq D$ and $1 \leq j \leq N - D$

$$\text{index}_1(\mathbf{t}, i) \geq \text{index}_1(\mathbf{s}, i) - 1 \text{ and } \text{index}_0(\mathbf{t}, j) \geq \text{index}_0(\mathbf{s}, j) - 1$$

That is, the ith copy of $x \in \{0, 1\}$ can only move to the left at most one position between $\mathbf{s}$ and $\mathbf{t}$. This condition is satisfied iff $\mathbf{s}$ can be transformed into $\mathbf{t}$ by non-overlapping right-shifts.

Unfortunately, the condition is not sufficient. For example, the following is a *(circular) right-shift Gray code* for $\mathbb{N}(7, 3)$

$$0000\overrightarrow{111}, 000\overrightarrow{1}101, 0010\overrightarrow{10}1, 00\overrightarrow{10}011, 000\overrightarrow{1}011$$

Therefore, the (prime-prefix) concatenation gives

$$0000111000110100101010010011000 1011$$

This is not a fixed-density de Bruijn cycle since $010010 \in \mathbb{B}'(7, 3)$ appears twice. Therefore, a special right-shift Gray code will be required.

Necessary Condition using Right-Shifts

If $\mathbf{s}, \mathbf{t} \in \mathbb{B}(N)$ then $\mathbf{s} \cdot \mathbf{t}$ can appear as a substring of a fixed-density de Bruijn cycle for $\mathbb{N}(N, D)$ only if for all $1 \leq i \leq D$ and $1 \leq j \leq N - D$

$$\text{index}_1(\mathbf{t}, i) \geq \text{index}_1(\mathbf{s}, i) - 1 \text{ and } \text{index}_0(\mathbf{t}, j) \geq \text{index}_0(\mathbf{s}, j) - 1$$

That is, the ith copy of $x \in \{0, 1\}$ can only move to the left at most one position between $\mathbf{s}$ and $\mathbf{t}$. This condition is satisfied iff $\mathbf{s}$ can be transformed into $\mathbf{t}$ by non-overlapping right-shifts.

Unfortunately, the condition is not sufficient. For example, the following is a *(circular) right-shift Gray code* for $\mathbb{N}(7, 3)$

$$0000\overrightarrow{111}, 000\overrightarrow{1}101, 0010\overrightarrow{10}1, 00\overrightarrow{10}011, 000\overrightarrow{1}011$$

Therefore, the (prime-prefix) concatenation gives

$$000011100011010010101001100011011$$

This is not a fixed-density de Bruijn cycle since $010010 \in \mathbb{B}'(7, 3)$ appears twice. Therefore, a special right-shift Gray code will be required.

Necessary Condition using Right-Shifts

If $\mathbf{s}, \mathbf{t} \in \mathbb{B}(N)$ then $\mathbf{s} \cdot \mathbf{t}$ can appear as a substring of a fixed-density de Bruijn cycle for $\mathbb{N}(N, D)$ only if for all $1 \leq i \leq D$ and $1 \leq j \leq N - D$

$$\text{index}_1(\mathbf{t}, i) \geq \text{index}_1(\mathbf{s}, i) - 1 \text{ and } \text{index}_0(\mathbf{t}, j) \geq \text{index}_0(\mathbf{s}, j) - 1$$

That is, the ith copy of $x \in \{0, 1\}$ can only move to the left at most one position between $\mathbf{s}$ and $\mathbf{t}$. This condition is satisfied iff $\mathbf{s}$ can be transformed into $\mathbf{t}$ by non-overlapping right-shifts.

Unfortunately, the condition is not sufficient. For example, the following is a *(circular) right-shift Gray code* for $\mathbb{N}(7, 3)$

$$0000\overrightarrow{111}, 000\overrightarrow{1}101, 0010\overrightarrow{10}1, 00\overrightarrow{10}011, 000\overrightarrow{1}011$$

Therefore, the (prime-prefix) concatenation gives

$$0000111000110100101010010011000 1011$$

This is not a fixed-density de Bruijn cycle since $010010 \in \mathbb{B}'(7, 3)$ appears twice. Therefore, a special right-shift Gray code will be required.

Necessary Condition using Right-Shifts

If $\mathbf{s}, \mathbf{t} \in \mathbb{B}(N)$ then $\mathbf{s} \cdot \mathbf{t}$ can appear as a substring of a fixed-density de Bruijn cycle for $\mathbb{N}(N, D)$ only if for all $1 \leq i \leq D$ and $1 \leq j \leq N - D$

$$\text{index}_1(\mathbf{t}, i) \geq \text{index}_1(\mathbf{s}, i) - 1 \text{ and } \text{index}_0(\mathbf{t}, j) \geq \text{index}_0(\mathbf{s}, j) - 1$$

That is, the ith copy of $x \in \{0, 1\}$ can only move to the left at most one position between $\mathbf{s}$ and $\mathbf{t}$. This condition is satisfied iff $\mathbf{s}$ can be transformed into $\mathbf{t}$ by non-overlapping right-shifts.

Unfortunately, the condition is not sufficient. For example, the following is a *(circular) right-shift Gray code* for $\mathbb{N}(7, 3)$

$$0000\overrightarrow{11}1, 000\overrightarrow{1}101, 0010\overrightarrow{10}1, 00\overrightarrow{10}011, 000\overrightarrow{1}011$$

Therefore, the (prime-prefix) concatenation gives

$$0000111000110100101010010011000 1011$$

This is not a fixed-density de Bruijn cycle since $010010 \in \mathbb{B}'(7, 3)$ appears twice. Therefore, a special right-shift Gray code will be required.

Necessary Condition using Right-Shifts

If $\mathbf{s}, \mathbf{t} \in \mathbb{B}(N)$ then $\mathbf{s} \cdot \mathbf{t}$ can appear as a substring of a fixed-density de Bruijn cycle for $\mathbb{N}(N, D)$ only if for all $1 \leq i \leq D$ and $1 \leq j \leq N - D$

$$\text{index}_1(\mathbf{t}, i) \geq \text{index}_1(\mathbf{s}, i) - 1 \text{ and } \text{index}_0(\mathbf{t}, j) \geq \text{index}_0(\mathbf{s}, j) - 1$$

That is, the ith copy of $x \in \{0, 1\}$ can only move to the left at most one position between $\mathbf{s}$ and $\mathbf{t}$. This condition is satisfied iff $\mathbf{s}$ can be transformed into $\mathbf{t}$ by non-overlapping right-shifts.

Unfortunately, the condition is not sufficient. For example, the following is a *(circular) right-shift Gray code* for $\mathbb{N}(7, 3)$

$$0000\overrightarrow{111}, 000\overrightarrow{1}101, 0010\overrightarrow{10}1, 00\overrightarrow{10}011, 000\overrightarrow{1}011$$

Therefore, the (prime-prefix) concatenation gives

$$0000111000110100101010010011000 1011$$

This is not a fixed-density de Bruijn cycle since $010010 \in \mathbb{B}'(7, 3)$ appears twice. Therefore, a special right-shift Gray code will be required.

Necessary Condition using Right-Shifts

If $\mathbf{s}, \mathbf{t} \in \mathbb{B}(N)$ then $\mathbf{s} \cdot \mathbf{t}$ can appear as a substring of a fixed-density de Bruijn cycle for $\mathbb{N}(N, D)$ only if for all $1 \leq i \leq D$ and $1 \leq j \leq N - D$

$$\text{index}_1(\mathbf{t}, i) \geq \text{index}_1(\mathbf{s}, i) - 1 \text{ and } \text{index}_0(\mathbf{t}, j) \geq \text{index}_0(\mathbf{s}, j) - 1$$

That is, the ith copy of $x \in \{0, 1\}$ can only move to the left at most one position between $\mathbf{s}$ and $\mathbf{t}$. This condition is satisfied iff $\mathbf{s}$ can be transformed into $\mathbf{t}$ by non-overlapping right-shifts.

Unfortunately, the condition is not sufficient. For example, the following is a *(circular) right-shift Gray code* for $\mathbb{N}(7, 3)$

$$0000\overrightarrow{111}, 000\overrightarrow{1}101, 0010\overrightarrow{101}, 00\overrightarrow{10}011, 000\overrightarrow{1}011$$

Therefore, the (prime-prefix) concatenation gives

$$00001110001101001010100110001011$$

This is not a fixed-density de Bruijn cycle since $010010 \in \mathbb{B}'(7, 3)$ appears twice. Therefore, a special right-shift Gray code will be required.

Necessary Condition using Right-Shifts

If $\mathbf{s}, \mathbf{t} \in \mathbb{B}(N)$ then $\mathbf{s} \cdot \mathbf{t}$ can appear as a substring of a fixed-density de Bruijn cycle for $\mathbb{N}(N, D)$ only if for all $1 \leq i \leq D$ and $1 \leq j \leq N - D$

$$\text{index}_1(\mathbf{t}, i) \geq \text{index}_1(\mathbf{s}, i) - 1 \text{ and } \text{index}_0(\mathbf{t}, j) \geq \text{index}_0(\mathbf{s}, j) - 1$$

That is, the ith copy of $x \in \{0, 1\}$ can only move to the left at most one position between $\mathbf{s}$ and $\mathbf{t}$. This condition is satisfied iff $\mathbf{s}$ can be transformed into $\mathbf{t}$ by non-overlapping right-shifts.

Unfortunately, the condition is not sufficient. For example, the following is a *(circular) right-shift Gray code* for $\mathbb{N}(7, 3)$

$$0000\overrightarrow{111}, 000\overrightarrow{1}101, 0010\overrightarrow{101}, 00\overrightarrow{10}011, 000\overrightarrow{1}011$$

Therefore, the (prime-prefix) concatenation gives

$$0000111000110100101010010011000 1011$$

This is not a fixed-density de Bruijn cycle since $010010 \in \mathbb{B}'(7, 3)$ appears twice. Therefore, a special right-shift Gray code will be required.

Necessary Condition using Right-Shifts

If $\mathbf{s}, \mathbf{t} \in \mathbb{B}(N)$ then $\mathbf{s} \cdot \mathbf{t}$ can appear as a substring of a fixed-density de Bruijn cycle for $\mathbb{N}(N, D)$ only if for all $1 \leq i \leq D$ and $1 \leq j \leq N - D$

$$\text{index}_1(\mathbf{t}, i) \geq \text{index}_1(\mathbf{s}, i) - 1 \text{ and } \text{index}_0(\mathbf{t}, j) \geq \text{index}_0(\mathbf{s}, j) - 1$$

That is, the ith copy of $x \in \{0, 1\}$ can only move to the left at most one position between $\mathbf{s}$ and $\mathbf{t}$. This condition is satisfied iff $\mathbf{s}$ can be transformed into $\mathbf{t}$ by non-overlapping right-shifts.

Unfortunately, the condition is not sufficient. For example, the following is a *(circular) right-shift Gray code* for $\mathbb{N}(7, 3)$

$$0000\overrightarrow{11}, 000\overrightarrow{1}101, 0010\overrightarrow{10}1, 00\overrightarrow{10}011, 000\overrightarrow{1}011$$

Therefore, the (prime-prefix) concatenation gives

$$0000111000110100101010010011000 1011$$

This is not a fixed-density de Bruijn cycle since $010010 \in \mathbb{B}'(7, 3)$ appears twice. Therefore, a special right-shift Gray code will be required.

Necessary Condition using Right-Shifts

If $\mathbf{s}, \mathbf{t} \in \mathbb{B}(N)$ then $\mathbf{s} \cdot \mathbf{t}$ can appear as a substring of a fixed-density de Bruijn cycle for $\mathbb{N}(N, D)$ only if for all $1 \leq i \leq D$ and $1 \leq j \leq N - D$

$$\text{index}_1(\mathbf{t}, i) \geq \text{index}_1(\mathbf{s}, i) - 1 \text{ and } \text{index}_0(\mathbf{t}, j) \geq \text{index}_0(\mathbf{s}, j) - 1$$

That is, the ith copy of $x \in \{0, 1\}$ can only move to the left at most one position between $\mathbf{s}$ and $\mathbf{t}$. This condition is satisfied iff $\mathbf{s}$ can be transformed into $\mathbf{t}$ by non-overlapping right-shifts.

Unfortunately, the condition is not sufficient. For example, the following is a *(circular) right-shift Gray code* for $\mathbb{N}(7, 3)$

$$0000\overrightarrow{11}1, 000\overrightarrow{11}01, 0010\overrightarrow{10}1, 00\overrightarrow{10}011, 000\overrightarrow{10}11$$

Therefore, the (prime-prefix) concatenation gives

$$00001110001101001010100100110001011$$

This is not a fixed-density de Bruijn cycle since $010010 \in \mathbb{B}'(7, 3)$ appears twice. Therefore, a special right-shift Gray code will be required.

Necessary Condition using Right-Shifts

If $\mathbf{s}, \mathbf{t} \in \mathbb{B}(N)$ then $\mathbf{s} \cdot \mathbf{t}$ can appear as a substring of a fixed-density de Bruijn cycle for $\mathbb{N}(N, D)$ only if for all $1 \leq i \leq D$ and $1 \leq j \leq N - D$

$$\text{index}_1(\mathbf{t}, i) \geq \text{index}_1(\mathbf{s}, i) - 1 \text{ and } \text{index}_0(\mathbf{t}, j) \geq \text{index}_0(\mathbf{s}, j) - 1$$

That is, the ith copy of $x \in \{0, 1\}$ can only move to the left at most one position between $\mathbf{s}$ and $\mathbf{t}$. This condition is satisfied iff $\mathbf{s}$ can be transformed into $\mathbf{t}$ by non-overlapping right-shifts.

Unfortunately, the condition is not sufficient. For example, the following is a *(circular) right-shift Gray code* for $\mathbb{N}(7, 3)$

$$0000\overrightarrow{11}1, 000\overrightarrow{1}101, 0010\overrightarrow{10}1, 00\overrightarrow{10}011, 000\overrightarrow{1}011$$

Therefore, the (prime-prefix) concatenation gives

$$0000111000110100101010010011000 1011$$

This is not a fixed-density de Bruijn cycle since $010010 \in \mathbb{B}'(7, 3)$ appears twice. Therefore, a special right-shift Gray code will be required.

Shift Gray Code for Fixed-Density Binary Strings

The iterative rule below generates the *cool-lex* right-shift Gray code for $\mathbb{B}(N, D)$ denoted $\overrightarrow{\mathcal{C}}(\mathbb{B}(N, D))$ [Ruskey-W 2005].

> Right-shift the first bit until it passes over 10 or reaches the end of the string.

More formally, the *cool right-shift* for $\mathbf{s} \in \mathbb{B}(N, D)$ is

$$\overrightarrow{\text{cool}}(\mathbf{s}) = \begin{cases} \overrightarrow{\text{shift}}(\mathbf{s}, 1, N) & \text{if } k == N \\ \overrightarrow{\text{shift}}(\mathbf{s}, 1, k+1) & \text{otherwise } (k < N). \end{cases}$$

where k is the maximum value such that $s_2 s_2 \cdots s_k = 0^*1^*$.

Cool-lex order also gives right-shift Gray codes for balanced parentheses [Ruskey-W. 2008] and fixed-density necklaces $\overrightarrow{\mathcal{C}}(\mathbb{N}(N, D))$ [Sawada-W. 2009].

Recursively, cool-lex is similar to co-lexicographic order. Generalization to multiset permutations [W. 2009].

000111
$\xrightarrow{}$
001110
$\xrightarrow{}$
011100
$\xrightarrow{}$
111000
$\xrightarrow{}$
110100
$\xrightarrow{}$
101100
$\xrightarrow{}$
011010
$\xrightarrow{}$
110010
$\xrightarrow{}$
101010
$\xrightarrow{}$
010110
$\xrightarrow{}$
100110
$\xrightarrow{}$
001101
$\xrightarrow{}$
011001
$\xrightarrow{}$
110001
$\xrightarrow{}$
101001
$\xrightarrow{}$
010101
$\xrightarrow{}$
100101
$\xrightarrow{}$
001011
$\xrightarrow{}$
010011
$\xrightarrow{}$
100011
$\xrightarrow{}$

Shift Gray Code for Fixed-Density Binary Strings

The iterative rule below generates the *cool-lex* right-shift Gray code for $\mathbb{B}(N, D)$ denoted $\overrightarrow{\mathcal{C}}(\mathbb{B}(N, D))$ [Ruskey-W 2005].

> Right-shift the first bit until it passes over 10 or reaches the end of the string.

More formally, the *cool right-shift* for $\mathbf{s} \in \mathbb{B}(N, D)$ is

$$\overrightarrow{\text{cool}}(\mathbf{s}) = \begin{cases} \overrightarrow{\text{shift}}(\mathbf{s}, 1, N) & \text{if } k = N \\ \overrightarrow{\text{shift}}(\mathbf{s}, 1, k+1) & \text{otherwise } (k < N). \end{cases}$$

where k is the maximum value such that $s_2 s_2 \cdots s_k = 0^* 1^*$.

Cool-lex order also gives right-shift Gray codes for balanced parentheses [Ruskey-W. 2008] and fixed-density necklaces $\overrightarrow{\mathcal{C}}(\mathbb{N}(N, D))$ [Sawada-W. 2009].

Recursively, cool-lex is similar to co-lexicographic order. Generalization to multiset permutations [W. 2009].

$000111 \xrightarrow{}$
$001110 \xrightarrow{}$
$011100 \xrightarrow{}$
$111000 \xrightarrow{}$
$110100 \xrightarrow{}$
$101100 \xrightarrow{}$
$011010 \xrightarrow{}$
$110010 \xrightarrow{}$
$101010 \xrightarrow{}$
$010110 \xrightarrow{}$
$100110 \xrightarrow{}$
$001101 \xrightarrow{}$
$011001 \xrightarrow{}$
$110001 \xrightarrow{}$
$101001 \xrightarrow{}$
$010101 \xrightarrow{}$
$100101 \xrightarrow{}$
$001011 \xrightarrow{}$
$010011 \xrightarrow{}$
$100011 \xrightarrow{}$

Shift Gray Code for Fixed-Density Binary Strings

The iterative rule below generates the *cool-lex* right-shift Gray code for $\mathbb{B}(N, D)$ denoted $\overrightarrow{\mathcal{C}}(\mathbb{B}(N, D))$ [Ruskey-W 2005].

> Right-shift the first bit until it passes over 10 or reaches the end of the string.

More formally, the *cool right-shift* for $\mathbf{s} \in \mathbb{B}(N, D)$ is

$$\overrightarrow{\text{cool}}(\mathbf{s}) = \begin{cases} \overrightarrow{\text{shift}}(\mathbf{s}, 1, N) & \text{if } k = N \\ \overrightarrow{\text{shift}}(\mathbf{s}, 1, k+1) & \text{otherwise } (k < N). \end{cases}$$

where k is the maximum value such that $s_2 s_2 \cdots s_k = 0^*1^*$.

Cool-lex order also gives right-shift Gray codes for balanced parentheses [Ruskey-W. 2008] and fixed-density necklaces $\overrightarrow{\mathcal{C}}(\mathbb{N}(N, D))$ [Sawada-W. 2009].

Recursively, cool-lex is similar to co-lexicographic order. Generalization to multiset permutations [W. 2009].

$\underrightarrow{000111}$
$\underrightarrow{001110}$
$\underrightarrow{011100}$
$\underrightarrow{111000}$
$\underrightarrow{110100}$
$\underrightarrow{101100}$
$\underrightarrow{011010}$
$\underrightarrow{110010}$
$\underrightarrow{101010}$
$\underrightarrow{010110}$
$\underrightarrow{100110}$
$\underrightarrow{001101}$
$\underrightarrow{011001}$
$\underrightarrow{110001}$
$\underrightarrow{101001}$
$\underrightarrow{010101}$
$\underrightarrow{100101}$
$\underrightarrow{001011}$
$\underrightarrow{010011}$
$\underrightarrow{100011}$

Shift Gray Code for Fixed-Density Binary Strings

The iterative rule below generates the *cool-lex* right-shift Gray code for $\mathbb{B}(N, D)$ denoted $\overrightarrow{\mathcal{C}}(\mathbb{B}(N, D))$ [Ruskey-W 2005].

> Right-shift the first bit until it passes over 10 or reaches the end of the string.

More formally, the *cool right-shift* for $\mathbf{s} \in \mathbb{B}(N, D)$ is

$$\overrightarrow{\text{cool}}(\mathbf{s}) = \begin{cases} \overrightarrow{\text{shift}}(\mathbf{s}, 1, N) & \text{if } k = N \\ \overrightarrow{\text{shift}}(\mathbf{s}, 1, k+1) & \text{otherwise } (k < N). \end{cases}$$

where k is the maximum value such that $s_2 s_2 \cdots s_k = 0^*1^*$.

Cool-lex order also gives right-shift Gray codes for balanced parentheses [Ruskey-W. 2008] and fixed-density necklaces $\overrightarrow{\mathcal{C}}(\mathbb{N}(N, D))$ [Sawada-W. 2009].

Recursively, cool-lex is similar to co-lexicographic order. Generalization to multiset permutations [W. 2009].

$\xrightarrow{000111}$
$\xrightarrow{001110}$
$\xrightarrow{011100}$
$\xrightarrow{111000}$
$\xrightarrow{110100}$
$\xrightarrow{101100}$
$\xrightarrow{011010}$
$\xrightarrow{110010}$
$\xrightarrow{101010}$
$\xrightarrow{010110}$
$\xrightarrow{100110}$
$\xrightarrow{001101}$
$\xrightarrow{011001}$
$\xrightarrow{110001}$
$\xrightarrow{101001}$
$\xrightarrow{010101}$
$\xrightarrow{100101}$
$\xrightarrow{001011}$
$\xrightarrow{010011}$
$\xrightarrow{100011}$

Shift Gray Code for Fixed-Density Binary Strings

The iterative rule below generates the *cool-lex* right-shift Gray code for $\mathbb{B}(N, D)$ denoted $\overrightarrow{\mathcal{C}}(\mathbb{B}(N, D))$ [Ruskey-W 2005].

> Right-shift the first bit until it passes over 10 or reaches the end of the string.

More formally, the *cool right-shift* for $\mathbf{s} \in \mathbb{B}(N, D)$ is

$$\overrightarrow{\text{cool}}(\mathbf{s}) = \begin{cases} \overrightarrow{\text{shift}}(\mathbf{s}, 1, N) & \text{if } k = N \\ \overrightarrow{\text{shift}}(\mathbf{s}, 1, k+1) & \text{otherwise } (k < N). \end{cases}$$

where k is the maximum value such that $s_2 s_2 \cdots s_k = 0^*1^*$.

Cool-lex order also gives right-shift Gray codes for balanced parentheses [Ruskey-W. 2008] and fixed-density necklaces $\overrightarrow{\mathcal{C}}(\mathbb{N}(N, D))$ [Sawada-W. 2009].

Recursively, cool-lex is similar to co-lexicographic order. Generalization to multiset permutations [W. 2009].

$000111 \xrightarrow{}$
$001110 \xrightarrow{}$
$011100 \xrightarrow{}$
$111000 \xrightarrow{}$
$110100 \xrightarrow{}$
$101100 \xrightarrow{}$
$011010 \xrightarrow{}$
$110010 \xrightarrow{}$
$101010 \xrightarrow{}$
$010110 \xrightarrow{}$
$100110 \xrightarrow{}$
$001101 \xrightarrow{}$
$011001 \xrightarrow{}$
$110001 \xrightarrow{}$
$101001 \xrightarrow{}$
$010101 \xrightarrow{}$
$100101 \xrightarrow{}$
$001011 \xrightarrow{}$
$010011 \xrightarrow{}$
$100011 \xrightarrow{}$

Shift Gray Code for Fixed-Density Binary Strings

The iterative rule below generates the *cool-lex* right-shift Gray code for $\mathbb{B}(N, D)$ denoted $\overrightarrow{\mathcal{C}}(\mathbb{B}(N, D))$ [Ruskey-W 2005].

> Right-shift the first bit until it passes over 10 or reaches the end of the string.

More formally, the *cool right-shift* for $\mathbf{s} \in \mathbb{B}(N, D)$ is

$$\overrightarrow{\text{cool}}(\mathbf{s}) = \begin{cases} \overrightarrow{\text{shift}}(\mathbf{s}, 1, N) & \text{if } k = N \\ \overrightarrow{\text{shift}}(\mathbf{s}, 1, k+1) & \text{otherwise } (k < N). \end{cases}$$

where k is the maximum value such that $s_2 s_2 \cdots s_k = 0^*1^*$.

Cool-lex order also gives right-shift Gray codes for balanced parentheses [Ruskey-W. 2008] and fixed-density necklaces $\overrightarrow{\mathcal{C}}(\mathbb{N}(N, D))$ [Sawada-W. 2009].

Recursively, cool-lex is similar to co-lexicographic order. Generalization to multiset permutations [W. 2009].

$000111 \xrightarrow{}$
$001110 \xrightarrow{}$
$011100 \xrightarrow{}$
$111000 \xrightarrow{}$
$110100 \xrightarrow{}$
$101100 \xrightarrow{}$
$011010 \xrightarrow{}$
$110010 \xrightarrow{}$
$101010 \xrightarrow{}$
$010110 \xrightarrow{}$
$100110 \xrightarrow{}$
$001101 \xrightarrow{}$
$011001 \xrightarrow{}$
$110001 \xrightarrow{}$
$101001 \xrightarrow{}$
$010101 \xrightarrow{}$
$100101 \xrightarrow{}$
$001011 \xrightarrow{}$
$010011 \xrightarrow{}$
$100011 \xrightarrow{}$

Shift Gray Code for Fixed-Density Binary Strings

The iterative rule below generates the *cool-lex* right-shift Gray code for $\mathbb{B}(N, D)$ denoted $\overrightarrow{\mathcal{C}}(\mathbb{B}(N, D))$ [Ruskey-W 2005].

> Right-shift the first bit until it passes over 10 or reaches the end of the string.

More formally, the *cool right-shift* for $\mathbf{s} \in \mathbb{B}(N, D)$ is

$$\overrightarrow{\text{cool}}(\mathbf{s}) = \begin{cases} \overrightarrow{\text{shift}}(\mathbf{s}, 1, N) & \text{if } k = N \\ \overrightarrow{\text{shift}}(\mathbf{s}, 1, k+1) & \text{otherwise } (k < N). \end{cases}$$

where k is the maximum value such that $s_2 s_2 \cdots s_k = 0^*1^*$.

Cool-lex order also gives right-shift Gray codes for balanced parentheses [Ruskey-W. 2008] and fixed-density necklaces $\overrightarrow{\mathcal{C}}(\mathbb{N}(N, D))$ [Sawada-W. 2009].

Recursively, cool-lex is similar to co-lexicographic order. Generalization to multiset permutations [W. 2009].

$000111 \xrightarrow{}$
$001110 \xrightarrow{}$
$011100 \xrightarrow{}$
$111000 \xrightarrow{}$
$110100 \xrightarrow{}$
$101100 \xrightarrow{}$
$011010 \xrightarrow{}$
$110010 \xrightarrow{}$
$101010 \xrightarrow{}$
$010110 \xrightarrow{}$
$100110 \xrightarrow{}$
$001101 \xrightarrow{}$
$011001 \xrightarrow{}$
$110001 \xrightarrow{}$
$101001 \xrightarrow{}$
$010101 \xrightarrow{}$
$100101 \xrightarrow{}$
$001011 \xrightarrow{}$
$010011 \xrightarrow{}$
$100011 \xrightarrow{}$

Shift Gray Code for Fixed-Density Binary Strings

The iterative rule below generates the *cool-lex* right-shift Gray code for $\mathbb{B}(N, D)$ denoted $\overrightarrow{\mathcal{C}}(\mathbb{B}(N, D))$ [Ruskey-W 2005].

> Right-shift the first bit until it passes over 10 or reaches the end of the string.

More formally, the *cool right-shift* for $\mathbf{s} \in \mathbb{B}(N, D)$ is

$$\overrightarrow{\text{cool}}(\mathbf{s}) = \begin{cases} \overrightarrow{\text{shift}}(\mathbf{s}, 1, N) & \text{if } k = N \\ \overrightarrow{\text{shift}}(\mathbf{s}, 1, k+1) & \text{otherwise } (k < N). \end{cases}$$

where k is the maximum value such that $s_2 s_2 \cdots s_k = 0^*1^*$.

Cool-lex order also gives right-shift Gray codes for balanced parentheses [Ruskey-W. 2008] and fixed-density necklaces $\overrightarrow{\mathcal{C}}(\mathbb{N}(N, D))$ [Sawada-W. 2009].

Recursively, cool-lex is similar to co-lexicographic order. Generalization to multiset permutations [W. 2009].

$000111 \xrightarrow{}$
$001110 \xrightarrow{}$
$011100 \xrightarrow{}$
$111000 \xrightarrow{}$
$110100 \xrightarrow{}$
$101100 \xrightarrow{}$
$011010 \xrightarrow{}$
$110010 \xrightarrow{}$
$101010 \xrightarrow{}$
$010110 \xrightarrow{}$
$100110 \xrightarrow{}$
$001101 \xrightarrow{}$
$011001 \xrightarrow{}$
$110001 \xrightarrow{}$
$101001 \xrightarrow{}$
$010101 \xrightarrow{}$
$100101 \xrightarrow{}$
$001011 \xrightarrow{}$
$010011 \xrightarrow{}$
$100011 \xrightarrow{}$

Shift Gray Code for Fixed-Density Binary Strings

The iterative rule below generates the *cool-lex* right-shift Gray code for $\mathbb{B}(N, D)$ denoted $\overrightarrow{\mathcal{C}}(\mathbb{B}(N, D))$ [Ruskey-W 2005].

> Right-shift the first bit until it passes over 10 or reaches the end of the string.

More formally, the *cool right-shift* for $\mathbf{s} \in \mathbb{B}(N, D)$ is

$$\overrightarrow{\text{cool}}(\mathbf{s}) = \begin{cases} \overrightarrow{\text{shift}}(\mathbf{s}, 1, N) & \text{if } k = N \\ \overrightarrow{\text{shift}}(\mathbf{s}, 1, k+1) & \text{otherwise } (k < N). \end{cases}$$

where k is the maximum value such that $s_2 s_2 \cdots s_k = 0^*1^*$.

Cool-lex order also gives right-shift Gray codes for balanced parentheses [Ruskey-W. 2008] and fixed-density necklaces $\overrightarrow{\mathcal{C}}(\mathbb{N}(N, D))$ [Sawada-W. 2009].

Recursively, cool-lex is similar to co-lexicographic order. Generalization to multiset permutations [W. 2009].

$000111 \xrightarrow{}$
$001110 \xrightarrow{}$
$011100 \xrightarrow{}$
$111000 \xrightarrow{}$
$110100 \xrightarrow{}$
$101100 \xrightarrow{}$
$011010 \xrightarrow{}$
$110010 \xrightarrow{}$
$101010 \xrightarrow{}$
$010110 \xrightarrow{}$
$100110 \xrightarrow{}$
$001101 \xrightarrow{}$
$011001 \xrightarrow{}$
$110001 \xrightarrow{}$
$101001 \xrightarrow{}$
$010101 \xrightarrow{}$
$100101 \xrightarrow{}$
$001011 \xrightarrow{}$
$010011 \xrightarrow{}$
$100011 \xrightarrow{}$

Shift Gray Code for Fixed-Density Binary Strings

The iterative rule below generates the *cool-lex* right-shift Gray code for $\mathbb{B}(N, D)$ denoted $\overrightarrow{\mathcal{C}}(\mathbb{B}(N, D))$ [Ruskey-W 2005].

> Right-shift the first bit until it passes over 10 or reaches the end of the string.

More formally, the *cool right-shift* for $\mathbf{s} \in \mathbb{B}(N, D)$ is

$$\overrightarrow{\text{cool}}(\mathbf{s}) = \begin{cases} \overrightarrow{\text{shift}}(\mathbf{s}, 1, N) & \text{if } k = N \\ \overrightarrow{\text{shift}}(\mathbf{s}, 1, k+1) & \text{otherwise } (k < N). \end{cases}$$

where k is the maximum value such that $s_2 s_2 \cdots s_k = 0^*1^*$.

Cool-lex order also gives right-shift Gray codes for balanced parentheses [Ruskey-W. 2008] and fixed-density necklaces $\overrightarrow{\mathcal{C}}(\mathbb{N}(N, D))$ [Sawada-W. 2009].

Recursively, cool-lex is similar to co-lexicographic order. Generalization to multiset permutations [W. 2009].

$000111 \xrightarrow{}$
$001110 \xrightarrow{}$
$011100 \xrightarrow{}$
$111000 \xrightarrow{}$
$110100 \xrightarrow{}$
$101100 \xrightarrow{}$
$011010 \xrightarrow{}$
$110010 \xrightarrow{}$
$101010 \xrightarrow{}$
$010110 \xrightarrow{}$
$100110 \xrightarrow{}$
$001101 \xrightarrow{}$
$011001 \xrightarrow{}$
$110001 \xrightarrow{}$
$101001 \xrightarrow{}$
$010101 \xrightarrow{}$
$100101 \xrightarrow{}$
$001011 \xrightarrow{}$
$010011 \xrightarrow{}$
$100011 \xrightarrow{}$

Shift Gray Code for Fixed-Density Binary Strings

The iterative rule below generates the *cool-lex* right-shift Gray code for $\mathbb{B}(N, D)$ denoted $\overrightarrow{\mathcal{C}}(\mathbb{B}(N, D))$ [Ruskey-W 2005].

> Right-shift the first bit until it passes over 10 or reaches the end of the string.

More formally, the *cool right-shift* for $\mathbf{s} \in \mathbb{B}(N, D)$ is

$$\overrightarrow{\text{cool}}(\mathbf{s}) = \begin{cases} \overrightarrow{\text{shift}}(\mathbf{s}, 1, N) & \text{if } k = N \\ \overrightarrow{\text{shift}}(\mathbf{s}, 1, k+1) & \text{otherwise } (k < N). \end{cases}$$

where k is the maximum value such that $s_2 s_2 \cdots s_k = 0^*1^*$.

Cool-lex order also gives right-shift Gray codes for balanced parentheses [Ruskey-W. 2008] and fixed-density necklaces $\overrightarrow{\mathcal{C}}(\mathbb{N}(N, D))$ [Sawada-W. 2009].

Recursively, cool-lex is similar to co-lexicographic order. Generalization to multiset permutations [W. 2009].

$\xrightarrow{000111}$
$\xrightarrow{001110}$
$\xrightarrow{011100}$
$\xrightarrow{111000}$
$\xrightarrow{110100}$
$\xrightarrow{101100}$
$\xrightarrow{011010}$
$\xrightarrow{110010}$
$\xrightarrow{101010}$
$\xrightarrow{010110}$
$\xrightarrow{100110}$
$\xrightarrow{001101}$
$\xrightarrow{011001}$
$\xrightarrow{110001}$
$\xrightarrow{101001}$
$\xrightarrow{010101}$
$\xrightarrow{100101}$
$\xrightarrow{001011}$
$\xrightarrow{010011}$
$\xrightarrow{100011}$

Shift Gray Code for Fixed-Density Binary Strings

The iterative rule below generates the *cool-lex* right-shift Gray code for $\mathbb{B}(N, D)$ denoted $\overrightarrow{\mathcal{C}}(\mathbb{B}(N, D))$ [Ruskey-W 2005].

> Right-shift the first bit until it passes over 10 or reaches the end of the string.

More formally, the *cool right-shift* for $\mathbf{s} \in \mathbb{B}(N, D)$ is

$$\overrightarrow{\text{cool}}(\mathbf{s}) = \begin{cases} \overrightarrow{\text{shift}}(\mathbf{s}, 1, N) & \text{if } k = N \\ \overrightarrow{\text{shift}}(\mathbf{s}, 1, k+1) & \text{otherwise } (k < N). \end{cases}$$

where k is the maximum value such that $s_2 s_2 \cdots s_k = 0^*1^*$.

Cool-lex order also gives right-shift Gray codes for balanced parentheses [Ruskey-W. 2008] and fixed-density necklaces $\overrightarrow{\mathcal{C}}(\mathbb{N}(N, D))$ [Sawada-W. 2009].

Recursively, cool-lex is similar to co-lexicographic order. Generalization to multiset permutations [W. 2009].

$000111 \xrightarrow{}$
$001110 \xrightarrow{}$
$011100 \xrightarrow{}$
$111000 \xrightarrow{}$
$110100 \xrightarrow{}$
$101100 \xrightarrow{}$
$011010 \xrightarrow{}$
$110010 \xrightarrow{}$
$101010 \xrightarrow{}$
$010110 \xrightarrow{}$
$100110 \xrightarrow{}$
$001101 \xrightarrow{}$
$011001 \xrightarrow{}$
$110001 \xrightarrow{}$
$101001 \xrightarrow{}$
$010101 \xrightarrow{}$
$100101 \xrightarrow{}$
$001011 \xrightarrow{}$
$010011 \xrightarrow{}$
$100011 \xrightarrow{}$

Shift Gray Code for Fixed-Density Binary Strings

The iterative rule below generates the *cool-lex* right-shift Gray code for $\mathbb{B}(N, D)$ denoted $\overrightarrow{\mathcal{C}}(\mathbb{B}(N, D))$ [Ruskey-W 2005].

> Right-shift the first bit until it passes over 10 or reaches the end of the string.

More formally, the *cool right-shift* for $\mathbf{s} \in \mathbb{B}(N, D)$ is

$$\overrightarrow{\text{cool}}(\mathbf{s}) = \begin{cases} \overrightarrow{\text{shift}}(\mathbf{s}, 1, N) & \text{if } k = N \\ \overrightarrow{\text{shift}}(\mathbf{s}, 1, k+1) & \text{otherwise } (k < N). \end{cases}$$

where k is the maximum value such that $s_2 s_2 \cdots s_k = 0^*1^*$.

Cool-lex order also gives right-shift Gray codes for balanced parentheses [Ruskey-W. 2008] and fixed-density necklaces $\overrightarrow{\mathcal{C}}(\mathbb{N}(N, D))$ [Sawada-W. 2009].

Recursively, cool-lex is similar to co-lexicographic order. Generalization to multiset permutations [W. 2009].

$000111 \xrightarrow{}$
$001110 \xrightarrow{}$
$011100 \xrightarrow{}$
$111000 \xrightarrow{}$
$110100 \xrightarrow{}$
$101100 \xrightarrow{}$
$011010 \xrightarrow{}$
$110010 \xrightarrow{}$
$101010 \xrightarrow{}$
$010110 \xrightarrow{}$
$100110 \xrightarrow{}$
$001101 \xrightarrow{}$
$011001 \xrightarrow{}$
$110001 \xrightarrow{}$
$101001 \xrightarrow{}$
$010101 \xrightarrow{}$
$100101 \xrightarrow{}$
$001011 \xrightarrow{}$
$010011 \xrightarrow{}$
$100011 \xrightarrow{}$

Shift Gray Code for Fixed-Density Binary Strings

The iterative rule below generates the *cool-lex* right-shift Gray code for $\mathbb{B}(N, D)$ denoted $\overrightarrow{\mathcal{C}}(\mathbb{B}(N, D))$ [Ruskey-W 2005].

> Right-shift the first bit until it passes over 10 or reaches the end of the string.

More formally, the *cool right-shift* for $\mathbf{s} \in \mathbb{B}(N, D)$ is

$$\overrightarrow{\text{cool}}(\mathbf{s}) = \begin{cases} \overrightarrow{\text{shift}}(\mathbf{s}, 1, N) & \text{if } k = N \\ \overrightarrow{\text{shift}}(\mathbf{s}, 1, k+1) & \text{otherwise } (k < N). \end{cases}$$

where k is the maximum value such that $s_2 s_2 \cdots s_k = 0^*1^*$.

Cool-lex order also gives right-shift Gray codes for balanced parentheses [Ruskey-W. 2008] and fixed-density necklaces $\overrightarrow{\mathcal{C}}(\mathbb{N}(N, D))$ [Sawada-W. 2009].

Recursively, cool-lex is similar to co-lexicographic order. Generalization to multiset permutations [W. 2009].

$000111 \xrightarrow{}$
$001110 \xrightarrow{}$
$011100 \xrightarrow{}$
$111000 \xrightarrow{}$
$110100 \xrightarrow{}$
$101100 \xrightarrow{}$
$011010 \xrightarrow{}$
$110010 \xrightarrow{}$
$101010 \xrightarrow{}$
$010110 \xrightarrow{}$
$100110 \xrightarrow{}$
$001101 \xrightarrow{}$
$011001 \xrightarrow{}$
$110001 \xrightarrow{}$
$101001 \xrightarrow{}$
$010101 \xrightarrow{}$
$100101 \xrightarrow{}$
$001011 \xrightarrow{}$
$010011 \xrightarrow{}$
$100011 \xrightarrow{}$

Shift Gray Code for Fixed-Density Binary Strings

The iterative rule below generates the *cool-lex* right-shift Gray code for $\mathbb{B}(N, D)$ denoted $\overrightarrow{\mathcal{C}}(\mathbb{B}(N, D))$ [Ruskey-W 2005].

> Right-shift the first bit until it passes over 10 or reaches the end of the string.

More formally, the *cool right-shift* for $\mathbf{s} \in \mathbb{B}(N, D)$ is

$$\overrightarrow{\text{cool}}(\mathbf{s}) = \begin{cases} \overrightarrow{\text{shift}}(\mathbf{s}, 1, N) & \text{if } k = N \\ \overrightarrow{\text{shift}}(\mathbf{s}, 1, k+1) & \text{otherwise } (k < N). \end{cases}$$

where k is the maximum value such that $s_2 s_2 \cdots s_k = 0^*1^*$.

Cool-lex order also gives right-shift Gray codes for balanced parentheses [Ruskey-W. 2008] and fixed-density necklaces $\overrightarrow{\mathcal{C}}(\mathbb{N}(N, D))$ [Sawada-W. 2009].

Recursively, cool-lex is similar to co-lexicographic order. Generalization to multiset permutations [W. 2009].

$000111 \xrightarrow{}$
$001110 \xrightarrow{}$
$011100 \xrightarrow{}$
$111000 \xrightarrow{}$
$110100 \xrightarrow{}$
$101100 \xrightarrow{}$
$011010 \xrightarrow{}$
$110010 \xrightarrow{}$
$101010 \xrightarrow{}$
$010110 \xrightarrow{}$
$100110 \xrightarrow{}$
$001101 \xrightarrow{}$
$011001 \xrightarrow{}$
$110001 \xrightarrow{}$
$101001 \xrightarrow{}$
$010101 \xrightarrow{}$
$100101 \xrightarrow{}$
$001011 \xrightarrow{}$
$010011 \xrightarrow{}$
$100011 \xrightarrow{}$

Shift Gray Code for Fixed-Density Binary Strings

The iterative rule below generates the *cool-lex* right-shift Gray code for $\mathbb{B}(N, D)$ denoted $\overrightarrow{\mathcal{C}}(\mathbb{B}(N, D))$ [Ruskey-W 2005].

> Right-shift the first bit until it passes over 10 or reaches the end of the string.

More formally, the *cool right-shift* for $\mathbf{s} \in \mathbb{B}(N, D)$ is

$$\overrightarrow{\text{cool}}(\mathbf{s}) = \begin{cases} \overrightarrow{\text{shift}}(\mathbf{s}, 1, N) & \text{if } k = N \\ \overrightarrow{\text{shift}}(\mathbf{s}, 1, k+1) & \text{otherwise } (k < N). \end{cases}$$

where k is the maximum value such that $s_2 s_2 \cdots s_k = 0^*1^*$.

Cool-lex order also gives right-shift Gray codes for balanced parentheses [Ruskey-W. 2008] and fixed-density necklaces $\overrightarrow{\mathcal{C}}(\mathbb{N}(N, D))$ [Sawada-W. 2009].

Recursively, cool-lex is similar to co-lexicographic order. Generalization to multiset permutations [W. 2009].

$000111 \xrightarrow{}$
$001110 \xrightarrow{}$
$011100 \xrightarrow{}$
$111000 \xrightarrow{}$
$110100 \xrightarrow{}$
$101100 \xrightarrow{}$
$011010 \xrightarrow{}$
$110010 \xrightarrow{}$
$101010 \xrightarrow{}$
$010110 \xrightarrow{}$
$100110 \xrightarrow{}$
$001101 \xrightarrow{}$
$011001 \xrightarrow{}$
$110001 \xrightarrow{}$
$101001 \xrightarrow{}$
$010101 \xrightarrow{}$
$100101 \xrightarrow{}$
$001011 \xrightarrow{}$
$010011 \xrightarrow{}$
$100011 \xrightarrow{}$

Shift Gray Code for Fixed-Density Binary Strings

The iterative rule below generates the *cool-lex* right-shift Gray code for $\mathbb{B}(N, D)$ denoted $\overrightarrow{\mathcal{C}}(\mathbb{B}(N, D))$ [Ruskey-W 2005].

> Right-shift the first bit until it passes over 10 or reaches the end of the string.

More formally, the *cool right-shift* for $\mathbf{s} \in \mathbb{B}(N, D)$ is

$$\overrightarrow{\text{cool}}(\mathbf{s}) = \begin{cases} \overrightarrow{\text{shift}}(\mathbf{s}, 1, N) & \text{if } k = N \\ \overrightarrow{\text{shift}}(\mathbf{s}, 1, k+1) & \text{otherwise } (k < N). \end{cases}$$

where k is the maximum value such that $s_2 s_2 \cdots s_k = 0^*1^*$.

Cool-lex order also gives right-shift Gray codes for balanced parentheses [Ruskey-W. 2008] and fixed-density necklaces $\overrightarrow{\mathcal{C}}(\mathbb{N}(N, D))$ [Sawada-W. 2009].

Recursively, cool-lex is similar to co-lexicographic order. Generalization to multiset permutations [W. 2009].

$000111 \xrightarrow{}$
$001110 \xrightarrow{}$
$011100 \xrightarrow{}$
$111000 \xrightarrow{}$
$110100 \xrightarrow{}$
$101100 \xrightarrow{}$
$011010 \xrightarrow{}$
$110010 \xrightarrow{}$
$101010 \xrightarrow{}$
$010110 \xrightarrow{}$
$100110 \xrightarrow{}$
$001101 \xrightarrow{}$
$011001 \xrightarrow{}$
$110001 \xrightarrow{}$
$101001 \xrightarrow{}$
$010101 \xrightarrow{}$
$100101 \xrightarrow{}$
$001011 \xrightarrow{}$
$010011 \xrightarrow{}$
$100011 \xrightarrow{}$

Shift Gray Code for Fixed-Density Binary Strings

The iterative rule below generates the *cool-lex* right-shift Gray code for $\mathbb{B}(N, D)$ denoted $\overrightarrow{\mathcal{C}}(\mathbb{B}(N, D))$ [Ruskey-W 2005].

> Right-shift the first bit until it passes over 10 or reaches the end of the string.

More formally, the *cool right-shift* for $\mathbf{s} \in \mathbb{B}(N, D)$ is

$$\overrightarrow{\text{cool}}(\mathbf{s}) = \begin{cases} \overrightarrow{\text{shift}}(\mathbf{s}, 1, N) & \text{if } k = N \\ \overrightarrow{\text{shift}}(\mathbf{s}, 1, k+1) & \text{otherwise } (k < N). \end{cases}$$

where k is the maximum value such that $s_2 s_2 \cdots s_k = 0^*1^*$.

Cool-lex order also gives right-shift Gray codes for balanced parentheses [Ruskey-W. 2008] and fixed-density necklaces $\overrightarrow{\mathcal{C}}(\mathbb{N}(N, D))$ [Sawada-W. 2009].

Recursively, cool-lex is similar to co-lexicographic order. Generalization to multiset permutations [W. 2009].

$000111 \xrightarrow{}$
$001110 \xrightarrow{}$
$011100 \xrightarrow{}$
$111000 \xrightarrow{}$
$110100 \xrightarrow{}$
$101100 \xrightarrow{}$
$011010 \xrightarrow{}$
$110010 \xrightarrow{}$
$101010 \xrightarrow{}$
$010110 \xrightarrow{}$
$100110 \xrightarrow{}$
$001101 \xrightarrow{}$
$011001 \xrightarrow{}$
$110001 \xrightarrow{}$
$101001 \xrightarrow{}$
$010101 \xrightarrow{}$
$100101 \xrightarrow{}$
$001011 \xrightarrow{}$
$010011 \xrightarrow{}$
$100011 \xrightarrow{}$

Shift Gray Code for Fixed-Density Binary Strings

The iterative rule below generates the *cool-lex* right-shift Gray code for $\mathbb{B}(N, D)$ denoted $\overrightarrow{\mathcal{C}}(\mathbb{B}(N, D))$ [Ruskey-W 2005].

> Right-shift the first bit until it passes over 10 or reaches the end of the string.

More formally, the *cool right-shift* for $\mathbf{s} \in \mathbb{B}(N, D)$ is

$$\overrightarrow{\text{cool}}(\mathbf{s}) = \begin{cases} \overrightarrow{\text{shift}}(\mathbf{s}, 1, N) & \text{if } k = N \\ \overrightarrow{\text{shift}}(\mathbf{s}, 1, k+1) & \text{otherwise } (k < N). \end{cases}$$

where k is the maximum value such that $s_2 s_2 \cdots s_k = 0^*1^*$.

Cool-lex order also gives right-shift Gray codes for balanced parentheses [Ruskey-W. 2008] and fixed-density necklaces $\overrightarrow{\mathcal{C}}(\mathbb{N}(N, D))$ [Sawada-W. 2009].

Recursively, cool-lex is similar to co-lexicographic order. Generalization to multiset permutations [W. 2009].

$000111 \xrightarrow{}$
$001110 \xrightarrow{}$
$011100 \xrightarrow{}$
$111000 \xrightarrow{}$
$110100 \xrightarrow{}$
$101100 \xrightarrow{}$
$011010 \xrightarrow{}$
$110010 \xrightarrow{}$
$101010 \xrightarrow{}$
$010110 \xrightarrow{}$
$100110 \xrightarrow{}$
$001101 \xrightarrow{}$
$011001 \xrightarrow{}$
$110001 \xrightarrow{}$
$101001 \xrightarrow{}$
$010101 \xrightarrow{}$
$100101 \xrightarrow{}$
$001011 \xrightarrow{}$
$010011 \xrightarrow{}$
$100011 \xrightarrow{}$

Shift Gray Code for Fixed-Density Binary Strings

The iterative rule below generates the *cool-lex* right-shift Gray code for $\mathbb{B}(N, D)$ denoted $\overrightarrow{\mathcal{C}}(\mathbb{B}(N, D))$ [Ruskey-W 2005].

> Right-shift the first bit until it passes over 10 or reaches the end of the string.

More formally, the *cool right-shift* for $\mathbf{s} \in \mathbb{B}(N, D)$ is

$$\overrightarrow{\text{cool}}(\mathbf{s}) = \begin{cases} \overrightarrow{\text{shift}}(\mathbf{s}, 1, N) & \text{if } k = N \\ \overrightarrow{\text{shift}}(\mathbf{s}, 1, k+1) & \text{otherwise } (k < N). \end{cases}$$

where k is the maximum value such that $s_2 s_2 \cdots s_k = 0^*1^*$.

Cool-lex order also gives right-shift Gray codes for balanced parentheses [Ruskey-W. 2008] and fixed-density necklaces $\overrightarrow{\mathcal{C}}(\mathbb{N}(N, D))$ [Sawada-W. 2009].

Recursively, cool-lex is similar to co-lexicographic order. Generalization to multiset permutations [W. 2009].

$000111 \xrightarrow{}$
$001110 \xrightarrow{}$
$011100 \xrightarrow{}$
$111000 \xrightarrow{}$
$110100 \xrightarrow{}$
$101100 \xrightarrow{}$
$011010 \xrightarrow{}$
$110010 \xrightarrow{}$
$101010 \xrightarrow{}$
$010110 \xrightarrow{}$
$100110 \xrightarrow{}$
$001101 \xrightarrow{}$
$011001 \xrightarrow{}$
$110001 \xrightarrow{}$
$101001 \xrightarrow{}$
$010101 \xrightarrow{}$
$100101 \xrightarrow{}$
$001011 \xrightarrow{}$
$010011 \xrightarrow{}$
$100011 \xrightarrow{}$

Shift Gray Code for Fixed-Density Binary Strings

The iterative rule below generates the *cool-lex* right-shift Gray code for $\mathbb{B}(N, D)$ denoted $\overrightarrow{\mathcal{C}}(\mathbb{B}(N, D))$ [Ruskey-W 2005].

> Right-shift the first bit until it passes over 10 or reaches the end of the string.

More formally, the *cool right-shift* for $\mathbf{s} \in \mathbb{B}(N, D)$ is

$$\overrightarrow{\text{cool}}(\mathbf{s}) = \begin{cases} \overrightarrow{\text{shift}}(\mathbf{s}, 1, N) & \text{if } k = N \\ \overrightarrow{\text{shift}}(\mathbf{s}, 1, k+1) & \text{otherwise } (k < N). \end{cases}$$

where k is the maximum value such that $s_2 s_2 \cdots s_k = 0^*1^*$.

Cool-lex order also gives right-shift Gray codes for balanced parentheses [Ruskey-W. 2008] and fixed-density necklaces $\overrightarrow{\mathcal{C}}(\mathbb{N}(N, D))$ [Sawada-W. 2009].

Recursively, cool-lex is similar to co-lexicographic order. Generalization to multiset permutations [W. 2009].

$000111 \xrightarrow{}$
$001110 \xrightarrow{}$
$011100 \xrightarrow{}$
$111000 \xrightarrow{}$
$110100 \xrightarrow{}$
$101100 \xrightarrow{}$
$011010 \xrightarrow{}$
$110010 \xrightarrow{}$
$101010 \xrightarrow{}$
$010110 \xrightarrow{}$
$100110 \xrightarrow{}$
$001101 \xrightarrow{}$
$011001 \xrightarrow{}$
$110001 \xrightarrow{}$
$101001 \xrightarrow{}$
$010101 \xrightarrow{}$
$100101 \xrightarrow{}$
$001011 \xrightarrow{}$
$010011 \xrightarrow{}$
$100011 \xrightarrow{}$

Shift Gray Code for Fixed-Density Binary Strings

The iterative rule below generates the *cool-lex* right-shift Gray code for $\mathbb{B}(N, D)$ denoted $\overrightarrow{\mathcal{C}}(\mathbb{B}(N, D))$ [Ruskey-W 2005].

> Right-shift the first bit until it passes over 10 or reaches the end of the string.

More formally, the *cool right-shift* for $\mathbf{s} \in \mathbb{B}(N, D)$ is

$$\overrightarrow{\text{cool}}(\mathbf{s}) = \begin{cases} \overrightarrow{\text{shift}}(\mathbf{s}, 1, N) & \text{if } k = N \\ \overrightarrow{\text{shift}}(\mathbf{s}, 1, k+1) & \text{otherwise } (k < N). \end{cases}$$

where k is the maximum value such that $s_2 s_2 \cdots s_k = 0^* 1^*$.

Cool-lex order also gives right-shift Gray codes for balanced parentheses [Ruskey-W. 2008] and fixed-density necklaces $\overrightarrow{\mathcal{C}}(\mathbb{N}(N, D))$ [Sawada-W. 2009].

Recursively, cool-lex is similar to co-lexicographic order. Generalization to multiset permutations [W. 2009].

$\xrightarrow{000111}$
$\xrightarrow{001110}$
$\xrightarrow{011100}$
$\xrightarrow{111000}$
$\xrightarrow{110100}$
$\xrightarrow{101100}$
$\xrightarrow{011010}$
$\xrightarrow{110010}$
$\xrightarrow{101010}$
$\xrightarrow{010110}$
$\xrightarrow{100110}$
$\xrightarrow{001101}$
$\xrightarrow{011001}$
$\xrightarrow{110001}$
$\xrightarrow{101001}$
$\xrightarrow{010101}$
$\xrightarrow{100101}$
$\xrightarrow{001011}$
$\xrightarrow{010011}$
$\xrightarrow{100011}$

Shift Gray Code for Fixed-Density Binary Strings

The iterative rule below generates the *cool-lex* right-shift Gray code for $\mathbb{B}(N, D)$ denoted $\overrightarrow{\mathcal{C}}(\mathbb{B}(N, D))$ [Ruskey-W 2005].

> Right-shift the first bit until it passes over 10 or reaches the end of the string.

More formally, the *cool right-shift* for $\mathbf{s} \in \mathbb{B}(N, D)$ is

$$\overrightarrow{\text{cool}}(\mathbf{s}) = \begin{cases} \overrightarrow{\text{shift}}(\mathbf{s}, 1, N) & \text{if } k = N \\ \overrightarrow{\text{shift}}(\mathbf{s}, 1, k+1) & \text{otherwise } (k < N). \end{cases}$$

where k is the maximum value such that $s_2 s_2 \cdots s_k = 0^*1^*$.

Cool-lex order also gives right-shift Gray codes for balanced parentheses [Ruskey-W. 2008] and fixed-density necklaces $\overrightarrow{\mathcal{C}}(\mathbb{N}(N, D))$ [Sawada-W. 2009].

Recursively, cool-lex is similar to co-lexicographic order. Generalization to multiset permutations [W. 2009].

$\xrightarrow{000111}$
$\xrightarrow{001110}$
$\xrightarrow{011100}$
$\xrightarrow{111000}$
$\xrightarrow{110100}$
$\xrightarrow{101100}$
$\xrightarrow{011010}$
$\xrightarrow{110010}$
$\xrightarrow{101010}$
$\xrightarrow{010110}$
$\xrightarrow{100110}$
$\xrightarrow{001101}$
$\xrightarrow{011001}$
$\xrightarrow{110001}$
$\xrightarrow{101001}$
$\xrightarrow{010101}$
$\xrightarrow{100101}$
$\xrightarrow{001011}$
$\xrightarrow{010011}$
$\xrightarrow{100011}$

Shift Gray Code for Fixed-Density Binary Strings

The iterative rule below generates the *cool-lex* right-shift Gray code for $\mathbb{B}(N, D)$ denoted $\overrightarrow{\mathcal{C}}(\mathbb{B}(N, D))$ [Ruskey-W 2005].

> Right-shift the first bit until it passes over 10 or reaches the end of the string.

More formally, the *cool right-shift* for $\mathbf{s} \in \mathbb{B}(N, D)$ is

$$\overrightarrow{\text{cool}}(\mathbf{s}) = \begin{cases} \overrightarrow{\text{shift}}(\mathbf{s}, 1, N) & \text{if } k = N \\ \overrightarrow{\text{shift}}(\mathbf{s}, 1, k+1) & \text{otherwise } (k < N). \end{cases}$$

where k is the maximum value such that $s_2 s_2 \cdots s_k = 0^*1^*$.

Cool-lex order also gives right-shift Gray codes for balanced parentheses [Ruskey-W. 2008] and fixed-density necklaces $\overrightarrow{\mathcal{C}}(\mathbb{N}(N, D))$ [Sawada-W. 2009].

Recursively, cool-lex is similar to co-lexicographic order. Generalization to multiset permutations [W. 2009].

$\xrightarrow{000111}$
$\xrightarrow{001110}$
$\xrightarrow{011100}$
$\xrightarrow{111000}$
$\xrightarrow{110100}$
$\xrightarrow{101100}$
$\xrightarrow{011010}$
$\xrightarrow{110010}$
$\xrightarrow{101010}$
$\xrightarrow{010110}$
$\xrightarrow{100110}$
$\xrightarrow{001101}$
$\xrightarrow{011001}$
$\xrightarrow{110001}$
$\xrightarrow{101001}$
$\xrightarrow{010101}$
$\xrightarrow{100101}$
$\xrightarrow{001011}$
$\xrightarrow{010011}$
$\xrightarrow{100011}$

Shift Gray Code for Fixed-Density Binary Strings

The iterative rule below generates the *cool-lex* right-shift Gray code for $\mathbb{B}(N, D)$ denoted $\overrightarrow{\mathcal{C}}(\mathbb{B}(N, D))$ [Ruskey-W 2005].

> Right-shift the first bit until it passes over 10 or reaches the end of the string.

More formally, the *cool right-shift* for $\mathbf{s} \in \mathbb{B}(N, D)$ is

$$\overrightarrow{\text{cool}}(\mathbf{s}) = \begin{cases} \overrightarrow{\text{shift}}(\mathbf{s}, 1, N) & \text{if } k = N \\ \overrightarrow{\text{shift}}(\mathbf{s}, 1, k+1) & \text{otherwise } (k < N). \end{cases}$$

where k is the maximum value such that $s_2 s_2 \cdots s_k = 0^*1^*$.

Cool-lex order also gives right-shift Gray codes for balanced parentheses [Ruskey-W. 2008] and fixed-density necklaces $\overrightarrow{\mathcal{C}}(\mathbb{N}(N, D))$ [Sawada-W. 2009].

Recursively, cool-lex is similar to co-lexicographic order. Generalization to multiset permutations [W. 2009].

$\xrightarrow{000111}$
$\xrightarrow{001110}$
$\xrightarrow{011100}$
$\xrightarrow{111000}$
$\xrightarrow{110100}$
$\xrightarrow{101100}$
$\xrightarrow{011010}$
$\xrightarrow{110010}$
$\xrightarrow{101010}$
$\xrightarrow{010110}$
$\xrightarrow{100110}$
$\xrightarrow{001101}$
$\xrightarrow{011001}$
$\xrightarrow{110001}$
$\xrightarrow{101001}$
$\xrightarrow{010101}$
$\xrightarrow{100101}$
$\xrightarrow{001011}$
$\xrightarrow{010011}$
$\xrightarrow{100011}$

Shift Gray Code for Fixed-Density Binary Strings

The iterative rule below generates the *cool-lex* right-shift Gray code for $\mathbb{B}(N, D)$ denoted $\overrightarrow{\mathcal{C}}(\mathbb{B}(N, D))$ [Ruskey-W 2005].

> Right-shift the first bit until it passes over 10 or reaches the end of the string.

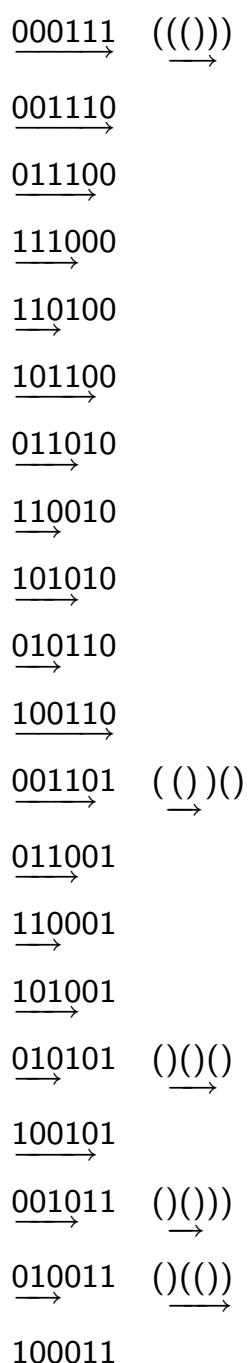
More formally, the *cool right-shift* for $\mathbf{s} \in \mathbb{B}(N, D)$ is

$$\overrightarrow{\text{cool}}(\mathbf{s}) = \begin{cases} \overrightarrow{\text{shift}}(\mathbf{s}, 1, N) & \text{if } k = N \\ \overrightarrow{\text{shift}}(\mathbf{s}, 1, k+1) & \text{otherwise } (k < N). \end{cases}$$

where k is the maximum value such that $s_2 s_2 \cdots s_k = 0^*1^*$.

Cool-lex order also gives right-shift Gray codes for balanced parentheses [Ruskey-W. 2008] and fixed-density necklaces $\overrightarrow{\mathcal{C}}(\mathbb{N}(N, D))$ [Sawada-W. 2009].

Recursively, cool-lex is similar to co-lexicographic order. Generalization to multiset permutations [W. 2009].



Shift Gray Code for Fixed-Density Binary Strings

The iterative rule below generates the *cool-lex* right-shift Gray code for $\mathbb{B}(N, D)$ denoted $\overrightarrow{\mathcal{C}}(\mathbb{B}(N, D))$ [Ruskey-W 2005].

> Right-shift the first bit until it passes over 10 or reaches the end of the string.

More formally, the *cool right-shift* for $\mathbf{s} \in \mathbb{B}(N, D)$ is

$$\overrightarrow{\text{cool}}(\mathbf{s}) = \begin{cases} \overrightarrow{\text{shift}}(\mathbf{s}, 1, N) & \text{if } k = N \\ \overrightarrow{\text{shift}}(\mathbf{s}, 1, k+1) & \text{otherwise } (k < N). \end{cases}$$

where k is the maximum value such that $s_2 s_2 \cdots s_k = 0^*1^*$.

Cool-lex order also gives right-shift Gray codes for balanced parentheses [Ruskey-W. 2008] and fixed-density necklaces $\overrightarrow{\mathcal{C}}(\mathbb{N}(N, D))$ [Sawada-W. 2009].

Recursively, cool-lex is similar to co-lexicographic order. Generalization to multiset permutations [W. 2009].

$\xrightarrow{000111}$
$\xrightarrow{001110}$
$\xrightarrow{011100}$
$\xrightarrow{111000}$
$\xrightarrow{110100}$
$\xrightarrow{101100}$
$\xrightarrow{011010}$
$\xrightarrow{110010}$
$\xrightarrow{101010}$
$\xrightarrow{010110}$
$\xrightarrow{100110}$
$\xrightarrow{001101}$
$\xrightarrow{011001}$
$\xrightarrow{110001}$
$\xrightarrow{101001}$
$\xrightarrow{010101}$
$\xrightarrow{100101}$
$\xrightarrow{001011}$
$\xrightarrow{010011}$
$\xrightarrow{100011}$

Shift Gray Code for Fixed-Density Binary Strings

The iterative rule below generates the *cool-lex* right-shift Gray code for $\mathbb{B}(N, D)$ denoted $\overrightarrow{\mathcal{C}}(\mathbb{B}(N, D))$ [Ruskey-W 2005].

> Right-shift the first bit until it passes over 10 or reaches the end of the string.

More formally, the *cool right-shift* for $\mathbf{s} \in \mathbb{B}(N, D)$ is

$$\overrightarrow{\text{cool}}(\mathbf{s}) = \begin{cases} \overrightarrow{\text{shift}}(\mathbf{s}, 1, N) & \text{if } k = N \\ \overrightarrow{\text{shift}}(\mathbf{s}, 1, k+1) & \text{otherwise } (k < N). \end{cases}$$

where k is the maximum value such that $s_2 s_2 \cdots s_k = 0^*1^*$.

Cool-lex order also gives right-shift Gray codes for balanced parentheses [Ruskey-W. 2008] and fixed-density necklaces $\overrightarrow{\mathcal{C}}(\mathbb{N}(N, D))$ [Sawada-W. 2009].

Recursively, cool-lex is similar to co-lexicographic order. Generalization to multiset permutations [W. 2009].

$\xrightarrow{000111}$
$\xrightarrow{001110}$
$\xrightarrow{011100}$
$\xrightarrow{111000}$
$\xrightarrow{110100}$
$\xrightarrow{101100}$
$\xrightarrow{011010}$
$\xrightarrow{110010}$
$\xrightarrow{101010}$
$\xrightarrow{010110}$
$\xrightarrow{100110}$
$\xrightarrow{001101}$
$\xrightarrow{011001}$
$\xrightarrow{110001}$
$\xrightarrow{101001}$
$\xrightarrow{010101}$
$\xrightarrow{100101}$
$\xrightarrow{001011}$
$\xrightarrow{010011}$
$\xrightarrow{100011}$

Shift Gray Code for Fixed-Density Binary Strings

The iterative rule below generates the *cool-lex* right-shift Gray code for $\mathbb{B}(N, D)$ denoted $\overrightarrow{\mathcal{C}}(\mathbb{B}(N, D))$ [Ruskey-W 2005].

> Right-shift the first bit until it passes over 10 or reaches the end of the string.

More formally, the *cool right-shift* for $\mathbf{s} \in \mathbb{B}(N, D)$ is

$$\overrightarrow{\text{cool}}(\mathbf{s}) = \begin{cases} \overrightarrow{\text{shift}}(\mathbf{s}, 1, N) & \text{if } k = N \\ \overrightarrow{\text{shift}}(\mathbf{s}, 1, k+1) & \text{otherwise } (k < N). \end{cases}$$

where k is the maximum value such that $s_2 s_2 \cdots s_k = 0^*1^*$.

Cool-lex order also gives right-shift Gray codes for balanced parentheses [Ruskey-W. 2008] and fixed-density necklaces $\overrightarrow{\mathcal{C}}(\mathbb{N}(N, D))$ [Sawada-W. 2009].

Recursively, cool-lex is similar to co-lexicographic order.

Generalization to multiset permutations [W. 2009].

$000111 \xrightarrow{}$
$001110 \xrightarrow{}$
$011100 \xrightarrow{}$
$111000 \xrightarrow{}$
$110100 \xrightarrow{}$
$101100 \xrightarrow{}$
$011010 \xrightarrow{}$
$110010 \xrightarrow{}$
$101010 \xrightarrow{}$
$010110 \xrightarrow{}$
$100110 \xrightarrow{}$
$001101 \xrightarrow{}$
$011001 \xrightarrow{}$
$110001 \xrightarrow{}$
$101001 \xrightarrow{}$
$010101 \xrightarrow{}$
$100101 \xrightarrow{}$
$001011 \xrightarrow{}$
$010011 \xrightarrow{}$
$100011 \xrightarrow{}$

Shift Gray Code for Fixed-Density Binary Strings

The iterative rule below generates the *cool-lex* right-shift Gray code for $\mathbb{B}(N, D)$ denoted $\overrightarrow{\mathcal{C}}(\mathbb{B}(N, D))$ [Ruskey-W 2005].

> Right-shift the first bit until it passes over 10 or reaches the end of the string.

More formally, the *cool right-shift* for $\mathbf{s} \in \mathbb{B}(N, D)$ is

$$\overrightarrow{\text{cool}}(\mathbf{s}) = \begin{cases} \overrightarrow{\text{shift}}(\mathbf{s}, 1, N) & \text{if } k = N \\ \overrightarrow{\text{shift}}(\mathbf{s}, 1, k+1) & \text{otherwise } (k < N). \end{cases}$$

where k is the maximum value such that $s_2 s_2 \cdots s_k = 0^*1^*$.

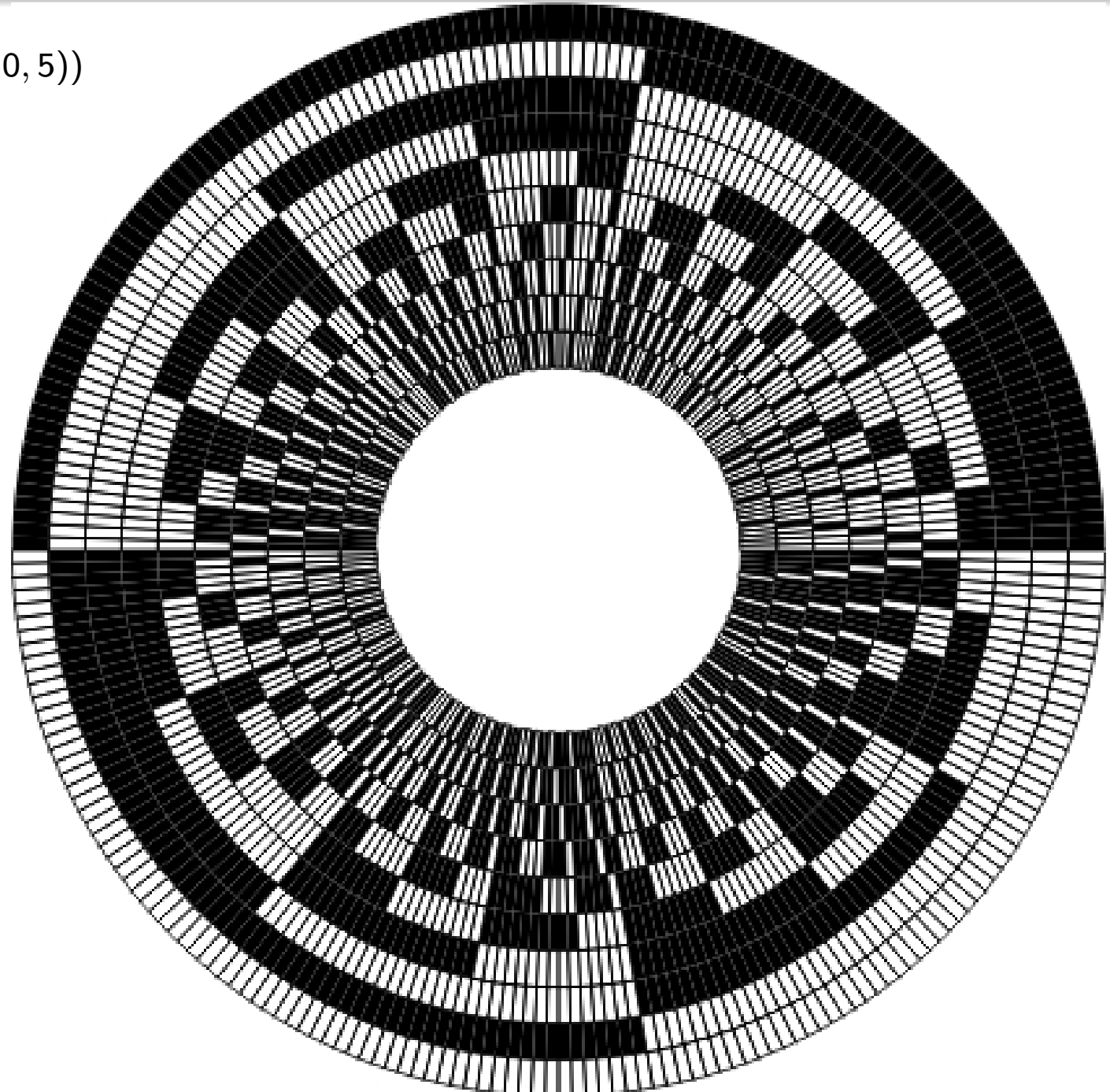
Cool-lex order also gives right-shift Gray codes for balanced parentheses [Ruskey-W. 2008] and fixed-density necklaces $\overrightarrow{\mathcal{C}}(\mathbb{N}(N, D))$ [Sawada-W. 2009].

Recursively, cool-lex is similar to co-lexicographic order. Generalization to multiset permutations [W. 2009].

$\xrightarrow{000111}$
$\xrightarrow{001110}$
$\xrightarrow{011100}$
$\xrightarrow{111000}$
$\xrightarrow{110100}$
$\xrightarrow{101100}$
$\xrightarrow{011010}$
$\xrightarrow{110010}$
$\xrightarrow{101010}$
$\xrightarrow{010110}$
$\xrightarrow{100110}$
$\xrightarrow{001101}$
$\xrightarrow{011001}$
$\xrightarrow{110001}$
$\xrightarrow{101001}$
$\xrightarrow{010101}$
$\xrightarrow{100101}$
$\xrightarrow{001011}$
$\xrightarrow{010011}$
$\xrightarrow{100011}$

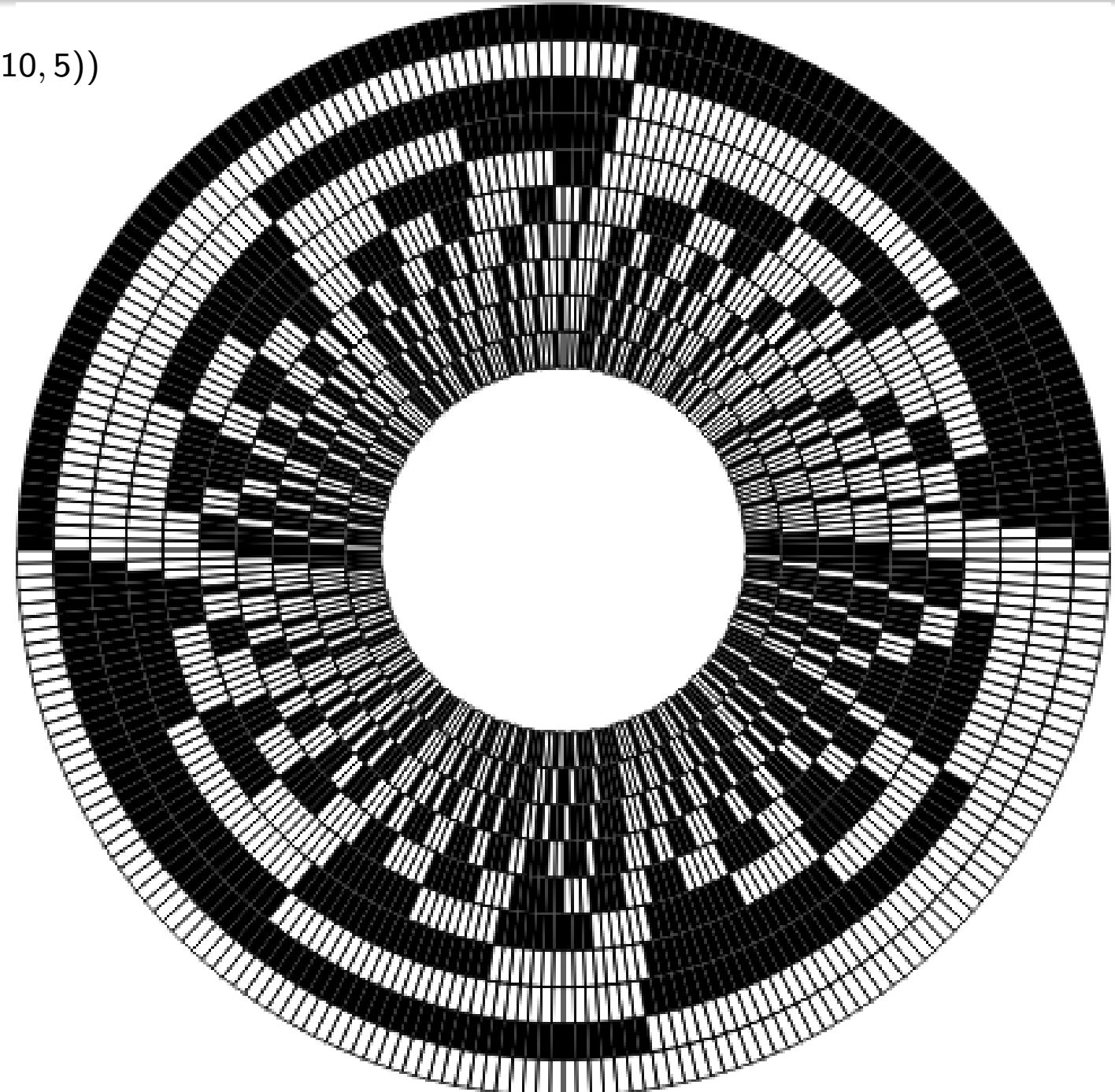
Lexicographic Order versus Cool-lex Order

$\mathcal{L}(\mathbb{B}(10,5))$



Lexicographic Order versus Cool-lex Order

$\overrightarrow{\mathcal{C}}(\mathbb{B}(10,5))$



Construction using Cool-lex Order

$\overrightarrow{C}(\mathbb{B}(6,3))$	$\overrightarrow{C}(\mathbb{N}(6,3))$	prime	·	$\mathbb{B}'(6,3)$	$\mathbb{B}(6,3)$
000111	000111	000111	0	00011	000111
001110			0	00111	001110
011100			0	01110	011100
111000			1	11100	111000
110100			1	11001	110010
101100			1	10011	100110
011010			0	00110	001101
110010			0	01101	011010
101010			1	11010	110100
010110			1	10101	101010
100110			0	01010	010101
001101	001101	001101	1	10100	101001
011001			0	01001	010011
110001			1	10010	100101
101001			0	00101	001011
010101	010101	01	0	01011	010110
100101			1	10110	101100
001011	001011	001011	0	01100	011001
010011			1	11000	110001
100011			1	10001	100011

This construction creates a fixed-density de Bruijn cycle for any $N \geq D \geq 0$.

Construction using Cool-lex Order

$\overrightarrow{C}(\mathbb{B}(6,3))$	$\overrightarrow{C}(\mathbb{N}(6,3))$	prime	$\cdot$	$\mathbb{B}'(6,3)$	$\mathbb{B}(6,3)$
000111	000111	000111	0	00011	000111
001110			0	00111	001110
011100			0	01110	011100
111000			1	11100	111000
110100			1	11001	110010
101100			1	10011	100110
011010			0	00110	001101
110010			0	01101	011010
101010			1	11010	110100
010110			1	10101	101010
100110			0	01010	010101
001101	001101	001101	1	10100	101001
011001			0	01001	010011
110001			1	10010	100101
101001			0	00101	001011
010101	010101	01	0	01011	010110
100101			1	10110	101100
001011	001011	001011	0	01100	011001
010011			1	11000	110001
100011			1	10001	100011

1. Start with the cool-lex order of binary strings of length N and density D.

Construction using Cool-lex Order

$\overrightarrow{C}(\mathbb{B}(6,3))$	$\overrightarrow{C}(\mathbb{N}(6,3))$	prime	·	$\mathbb{B}'(6,3)$	$\mathbb{B}(6,3)$
000111	000111	000111	0	00011	000111
001110			0	00111	001110
011100			0	01110	011100
111000			1	11100	111000
110100			1	11001	110010
101100			1	10011	100110
011010			0	00110	001101
110010			0	01101	011010
101010			1	11010	110100
010110			1	10101	101010
100110			0	01010	010101
001101	001101	001101	1	10100	101001
011001			0	01001	010011
110001			1	10010	100101
101001			0	00101	001011
010101	010101	01	0	01011	010110
100101			1	10110	101100
001011	001011	001011	0	01100	011001
010011			1	11000	110001
100011			1	10001	100011

2. Remove non-necklaces to get cool-lex of necklaces of length N and density D.

Construction using Cool-lex Order

$\overrightarrow{C}(\mathbb{B}(6,3))$	$\overrightarrow{C}(\mathbb{N}(6,3))$	prime	$\cdot$	$\mathbb{B}'(6,3)$	$\mathbb{B}(6,3)$
000111	000111	000111	0	00011	000111
001110			0	00111	001110
011100			0	01110	011100
111000			1	11100	111000
110100			1	11001	110010
101100			1	10011	100110
011010			0	00110	001101
110010			0	01101	011010
101010			1	11010	110100
010110			1	10101	101010
100110			0	01010	010101
001101	001101	001101	1	10100	101001
011001			0	01001	010011
110001			1	10010	100101
101001			0	00101	001011
010101	010101	01	0	01011	010110
100101			1	10110	101100
001011	001011	001011	0	01100	011001
010011			1	11000	110001
100011			1	10001	100011

3. Take the prime-prefix of each necklace.

Construction using Cool-lex Order

$\overrightarrow{C}(\mathbb{B}(6,3))$	$\overrightarrow{C}(\mathbb{N}(6,3))$	prime	$\cdot$	$\mathbb{B}'(6,3)$	$\mathbb{B}(6,3)$
000111	000111	000111	0	00011	000111
001110			0	00111	001110
011100			0	01110	011100
111000			1	11100	111000
110100			1	11001	110010
101100			1	10011	100110
011010			0	00110	001101
110010			0	01101	011010
101010			1	11010	110100
010110			1	10101	101010
100110			0	01010	010101
001101	001101	001101	1	10100	101001
011001			0	01001	010011
110001			1	10010	100101
101001			0	00101	001011
010101	010101	01	0	01011	010110
100101			1	10110	101100
001011	001011	001011	0	01100	011001
010011			1	11000	110001
100011			1	10001	100011

4. Concatenate the results to obtain the fixed-density de Bruijn cycle.

Construction using Cool-lex Order

$\overrightarrow{\mathcal{C}}(\mathbb{B}(6,3))$	$\overrightarrow{\mathcal{C}}(\mathbb{N}(6,3))$	prime	$\cdot$	$\mathbb{B}'(6,3)$	$\mathbb{B}(6,3)$
000111	000111	000111	0	00011	000111
001110			0	00111	001110
011100			0	01110	011100
111000			1	11100	111000
110100			1	11001	110010
101100			1	10011	100110
011010			0	00110	001101
110010			0	01101	011010
101010			1	11010	110100
010110			1	10101	101010
100110			0	01010	010101
001101	001101	001101	1	10100	101001
011001			0	01001	010011
110001			1	10010	100101
101001			0	00101	001011
010101	010101	01	0	01011	010110
100101			1	10110	101100
001011	001011	001011	0	01100	011001
010011			1	11000	110001
100011			1	10001	100011

(Correct length since $\circledcirc(\mathbf{s})$ contributes $|\circledcirc(\mathbf{s})|$ bits for each $\mathbf{s} \in \mathbb{B}(N, D)$.)

Construction using Cool-lex Order

$\overrightarrow{C}(\mathbb{B}(6,3))$	$\overrightarrow{C}(\mathbb{N}(6,3))$	prime	$\cdot$	$\mathbb{B}'(6,3)$	$\mathbb{B}(6,3)$
000111	000111	000111	0	00011	000111
001110			0	00111	001110
011100			0	01110	011100
111000			1	11100	111000
110100			1	11001	110010
101100			1	10011	100110
011010			0	00110	001101
110010			0	01101	011010
101010			1	11010	110100
010110			1	10101	101010
100110			0	01010	010101
001101	001101	001101	1	10100	101001
011001			0	01001	010011
110001			1	10010	100101
101001			0	00101	001011
010101	010101	01	0	01011	010110
100101			1	10110	101100
001011	001011	001011	0	01100	011001
010011			1	11000	110001
100011			1	10001	100011

Substrings equal $\mathbb{B}'(N, D)$.

Construction using Cool-lex Order

$\overrightarrow{C}(\mathbb{B}(6,3))$	$\overrightarrow{C}(\mathbb{N}(6,3))$	prime	·	$\mathbb{B}'(6,3)$	$\mathbb{B}(6,3)$
000111	000111	000111	0	00011	000111
001110			0	00111	001110
011100			0	01110	011100
111000			1	11100	111000
110100			1	11001	110010
101100			1	10011	100110
011010			0	00110	001101
110010			0	01101	011010
101010			1	11010	110100
010110			1	10101	101010
100110			0	01010	010101
001101	001101	001101	1	10100	101001
011001			0	01001	010011
110001			1	10010	100101
101001			0	00101	001011
010101	010101	01	0	01011	010110
100101			1	10110	101100
001011	001011	001011	0	01100	011001
010011			1	11000	110001
100011			1	10001	100011

Suffix the unique missing symbol to obtain $\mathbb{B}(N, D)$.

Construction using Cool-lex Order

$\overrightarrow{C}(\mathbb{B}(6,3))$	$\overrightarrow{C}(\mathbb{N}(6,3))$	prime	·	$\mathbb{B}'(6,3)$	$\mathbb{B}(6,3)$
000111	000111	000111	0	00011	000111
001110			0	00111	001110
011100			0	01110	011100
111000			1	11100	111000
110100			1	11001	110010
101100			1	10011	100110
011010			0	00110	001101
110010			0	01101	011010
101010			1	11010	110100
010110			1	10101	101010
100110			0	01010	010101
001101	001101	001101	1	10100	101001
011001			0	01001	010011
110001			1	10010	100101
101001			0	00101	001011
010101	010101	01	0	01011	010110
100101			1	10110	101100
001011	001011	001011	0	01100	011001
010011			1	11000	110001
100011			1	10001	100011

Successive fixed-density strings differ by prefix-shift of length N or $N-1$.

Construction using Cool-lex Order

$\overrightarrow{C}(\mathbb{B}(6,3))$	$\overrightarrow{C}(\mathbb{N}(6,3))$	prime	·	$\mathbb{B}'(6,3)$	$\mathbb{B}(6,3)$
000111	000111	000111	0	00011	000111
001110			0	00111	001110
011100			0	01110	011100
111000			1	11100	111000
110100			1	11001	110010
101100			1	10011	100110
011010			0	00110	001101
110010			0	01101	011010
101010			1	11010	110100
010110			1	10101	101010
100110			0	01010	010101
001101	001101	001101	1	10100	101001
011001			0	01001	010011
110001			1	10010	100101
101001			0	00101	001011
010101	010101	01	0	01011	010110
100101			1	10110	101100
001011	001011	001011	0	01100	011001
010011			1	11000	110001
100011			1	10001	100011

When $N = 2D$ the construction gives a universal cycle for the middle levels.

Final Thoughts

Summary

- Introduced concept of fixed-density de Bruijn cycle.
- Cool-lex order is the Grand-Daddy of fixed-density de Bruijn cycles as lexicographic order is the Grand-Daddy of de Bruijn cycles.

Open Problems

- Improve efficiency of this construction for fixed-density binary strings.
- Is this construction the first in lexicographic / cool-lex order?
- Maximize/minimize the number of 1s in a fixed-density de Bruijn cycle.
- Maximize/minimize the shifts of length N in resulting order of $\mathbb{B}(N, D)$.
- Existence of *density-range de Bruijn Cycles* for $\bigcup_{i=\min}^{\max} \mathbb{B}(N, i)$?
- Generalize to shorthand universal cycles for multiset permutations.
  - Binary multiset case solved in this talk.
  - Set permutation case solved [Ruskey-W. 2009] [Bell-ringers].

Acknowledgements

- Frank Ruskey and Joe Sawada
- Wendy Myrvold

Final Thoughts

Summary

- Introduced concept of fixed-density de Bruijn cycle.
- Cool-lex order is the Grand-Daddy of fixed-density de Bruijn cycles as lexicographic order is the Grand-Daddy of de Bruijn cycles.

Open Problems

- Improve efficiency of this construction for fixed-density binary strings.
- Is this construction the first in lexicographic / cool-lex order?
- Maximize/minimize the number of 1s in a fixed-density de Bruijn cycle.
- Maximize/minimize the shifts of length N in resulting order of $\mathbb{B}(N, D)$.
- Existence of *density-range de Bruijn Cycles* for $\bigcup_{i=\min}^{\max} \mathbb{B}(N, i)$?
- Generalize to shorthand universal cycles for multiset permutations.
  - Binary multiset case solved in this talk.
  - Set permutation case solved [Ruskey-W. 2009] [Bell-ringers].

Acknowledgements

- Frank Ruskey and Joe Sawada
- Wendy Myrvold

Final Thoughts

Summary

- Introduced concept of fixed-density de Bruijn cycle.
- Cool-lex order is the Grand-Daddy of fixed-density de Bruijn cycles as lexicographic order is the Grand-Daddy of de Bruijn cycles.

Open Problems

- Improve efficiency of this construction for fixed-density binary strings.
- Is this construction the first in lexicographic / cool-lex order?
- Maximize/minimize the number of 1s in a fixed-density de Bruijn cycle.
- Maximize/minimize the shifts of length N in resulting order of $\mathbb{B}(N, D)$.
- Existence of *density-range de Bruijn Cycles* for $\bigcup_{i=\min}^{\max} \mathbb{B}(N, i)$?
- Generalize to shorthand universal cycles for multiset permutations.
  - Binary multiset case solved in this talk.
  - Set permutation case solved [Ruskey-W. 2009] [Bell-ringers].

Acknowledgements

- Frank Ruskey and Joe Sawada
- Wendy Myrvold

Final Thoughts

Summary

- Introduced concept of fixed-density de Bruijn cycle.
- Cool-lex order is the Grand-Daddy of fixed-density de Bruijn cycles as lexicographic order is the Grand-Daddy of de Bruijn cycles.

Open Problems

- Improve efficiency of this construction for fixed-density binary strings.
- Is this construction the first in lexicographic / cool-lex order?
- Maximize/minimize the number of 1s in a fixed-density de Bruijn cycle.
- Maximize/minimize the shifts of length N in resulting order of $\mathbb{B}(N, D)$.
- Existence of *density-range de Bruijn Cycles* for $\bigcup_{i=\min}^{\max} \mathbb{B}(N, i)$?
- Generalize to shorthand universal cycles for multiset permutations.
  - Binary multiset case solved in this talk.
  - Set permutation case solved [Ruskey-W. 2009] [Bell-ringers].

Acknowledgements

- Frank Ruskey and Joe Sawada
- Wendy Myrvold

Final Thoughts

Summary

- Introduced concept of fixed-density de Bruijn cycle.
- Cool-lex order is the Grand-Daddy of fixed-density de Bruijn cycles as lexicographic order is the Grand-Daddy of de Bruijn cycles.

Open Problems

- Improve efficiency of this construction for fixed-density binary strings.
- Is this construction the first in lexicographic / cool-lex order?
- Maximize/minimize the number of 1s in a fixed-density de Bruijn cycle.
- Maximize/minimize the shifts of length N in resulting order of $\mathbb{B}(N, D)$.
- Existence of *density-range de Bruijn Cycles* for $\bigcup_{i=\min}^{\max} \mathbb{B}(N, i)$?
- Generalize to shorthand universal cycles for multiset permutations.
  - Binary multiset case solved in this talk.
  - Set permutation case solved [Ruskey-W. 2009] [Bell-ringers].

Acknowledgements

- Frank Ruskey and Joe Sawada
- Wendy Myrvold

Final Thoughts

Summary

- Introduced concept of fixed-density de Bruijn cycle.
- Cool-lex order is the Grand-Daddy of fixed-density de Bruijn cycles as lexicographic order is the Grand-Daddy of de Bruijn cycles.

Open Problems

- Improve efficiency of this construction for fixed-density binary strings.
- Is this construction the first in lexicographic / cool-lex order?
- Maximize/minimize the number of 1s in a fixed-density de Bruijn cycle.
- Maximize/minimize the shifts of length N in resulting order of $\mathbb{B}(N, D)$.
- Existence of *density-range de Bruijn Cycles* for $\bigcup_{i=\min}^{\max} \mathbb{B}(N, i)$?
- Generalize to shorthand universal cycles for multiset permutations.
  - Binary multiset case solved in this talk.
  - Set permutation case solved [Ruskey-W. 2009] [Bell-ringers].

Acknowledgements

- Frank Ruskey and Joe Sawada
- Wendy Myrvold

Final Thoughts

Summary

- Introduced concept of fixed-density de Bruijn cycle.
- Cool-lex order is the Grand-Daddy of fixed-density de Bruijn cycles as lexicographic order is the Grand-Daddy of de Bruijn cycles.

Open Problems

- Improve efficiency of this construction for fixed-density binary strings.
- Is this construction the first in lexicographic / cool-lex order?
- Maximize/minimize the number of 1s in a fixed-density de Bruijn cycle.
- Maximize/minimize the shifts of length N in resulting order of $\mathbb{B}(N, D)$.
- Existence of *density-range de Bruijn Cycles* for $\bigcup_{i=\min}^{\max} \mathbb{B}(N, i)$?
- Generalize to shorthand universal cycles for multiset permutations.
  - Binary multiset case solved in this talk.
  - Set permutation case solved [Ruskey-W. 2009] [Bell-ringers].

Acknowledgements

- Frank Ruskey and Joe Sawada
- Wendy Myrvold

Final Thoughts

Summary

- Introduced concept of fixed-density de Bruijn cycle.
- Cool-lex order is the Grand-Daddy of fixed-density de Bruijn cycles as lexicographic order is the Grand-Daddy of de Bruijn cycles.

Open Problems

- Improve efficiency of this construction for fixed-density binary strings.
- Is this construction the first in lexicographic / cool-lex order?
- Maximize/minimize the number of 1s in a fixed-density de Bruijn cycle.
- Maximize/minimize the shifts of length N in resulting order of $\mathbb{B}(N, D)$.
- Existence of *density-range de Bruijn Cycles* for $\bigcup_{i=\min}^{\max} \mathbb{B}(N, i)$?
- Generalize to shorthand universal cycles for multiset permutations.
  - Binary multiset case solved in this talk.
  - Set permutation case solved [Ruskey-W. 2009] [Bell-ringers].

Acknowledgements

- Frank Ruskey and Joe Sawada
- Wendy Myrvold

Final Thoughts

Summary

- Introduced concept of fixed-density de Bruijn cycle.
- Cool-lex order is the Grand-Daddy of fixed-density de Bruijn cycles as lexicographic order is the Grand-Daddy of de Bruijn cycles.

Open Problems

- Improve efficiency of this construction for fixed-density binary strings.
- Is this construction the first in lexicographic / cool-lex order?
- Maximize/minimize the number of 1s in a fixed-density de Bruijn cycle.
- Maximize/minimize the shifts of length N in resulting order of $\mathbb{B}(N, D)$.
- Existence of *density-range de Bruijn Cycles* for $\bigcup_{i=\min}^{\max} \mathbb{B}(N, i)$?
- Generalize to shorthand universal cycles for multiset permutations.
  - Binary multiset case solved in this talk.
  - Set permutation case solved [Ruskey-W. 2009] [Bell-ringers].

Acknowledgements

- Frank Ruskey and Joe Sawada
- Wendy Myrvold

Final Thoughts

Summary

- Introduced concept of fixed-density de Bruijn cycle.
- Cool-lex order is the Grand-Daddy of fixed-density de Bruijn cycles as lexicographic order is the Grand-Daddy of de Bruijn cycles.

Open Problems

- Improve efficiency of this construction for fixed-density binary strings.
- Is this construction the first in lexicographic / cool-lex order?
- Maximize/minimize the number of 1s in a fixed-density de Bruijn cycle.
- Maximize/minimize the shifts of length N in resulting order of $\mathbb{B}(N, D)$.
- Existence of *density-range de Bruijn Cycles* for $\bigcup_{i=\min}^{\max} \mathbb{B}(N, i)$?
- Generalize to shorthand universal cycles for multiset permutations.
  - Binary multiset case solved in this talk.
  - Set permutation case solved [Ruskey-W. 2009] [Bell-ringers].

Acknowledgements

- Frank Ruskey and Joe Sawada
- Wendy Myrvold

Final Thoughts

Summary

- Introduced concept of fixed-density de Bruijn cycle.
- Cool-lex order is the Grand-Daddy of fixed-density de Bruijn cycles as lexicographic order is the Grand-Daddy of de Bruijn cycles.

Open Problems

- Improve efficiency of this construction for fixed-density binary strings.
- Is this construction the first in lexicographic / cool-lex order?
- Maximize/minimize the number of 1s in a fixed-density de Bruijn cycle.
- Maximize/minimize the shifts of length N in resulting order of $\mathbb{B}(N, D)$.
- Existence of *density-range de Bruijn Cycles* for $\bigcup_{i=\min}^{\max} \mathbb{B}(N, i)$?
- Generalize to shorthand universal cycles for multiset permutations.
  - Binary multiset case solved in this talk.
  - Set permutation case solved [Ruskey-W. 2009] [Bell-ringers].

Acknowledgements

- Frank Ruskey and Joe Sawada
- Wendy Myrvold

Final Thoughts

Summary

- Introduced concept of fixed-density de Bruijn cycle.
- Cool-lex order is the Grand-Daddy of fixed-density de Bruijn cycles as lexicographic order is the Grand-Daddy of de Bruijn cycles.

Open Problems

- Improve efficiency of this construction for fixed-density binary strings.
- Is this construction the first in lexicographic / cool-lex order?
- Maximize/minimize the number of 1s in a fixed-density de Bruijn cycle.
- Maximize/minimize the shifts of length N in resulting order of $\mathbb{B}(N, D)$.
- Existence of *density-range de Bruijn Cycles* for $\bigcup_{i=\min}^{\max} \mathbb{B}(N, i)$?
- Generalize to shorthand universal cycles for multiset permutations.
  - Binary multiset case solved in this talk.
  - Set permutation case solved [Ruskey-W. 2009] [Bell-ringers].

Acknowledgements

- Frank Ruskey and Joe Sawada
- Wendy Myrvold

Final Thoughts

Summary

- Introduced concept of fixed-density de Bruijn cycle.
- Cool-lex order is the Grand-Daddy of fixed-density de Bruijn cycles as lexicographic order is the Grand-Daddy of de Bruijn cycles.

Open Problems

- Improve efficiency of this construction for fixed-density binary strings.
- Is this construction the first in lexicographic / cool-lex order?
- Maximize/minimize the number of 1s in a fixed-density de Bruijn cycle.
- Maximize/minimize the shifts of length N in resulting order of $\mathbb{B}(N, D)$.
- Existence of *density-range de Bruijn Cycles* for $\bigcup_{i=\min}^{\max} \mathbb{B}(N, i)$?
- Generalize to shorthand universal cycles for multiset permutations.
  - Binary multiset case solved in this talk.
  - Set permutation case solved [Ruskey-W. 2009] [Bell-ringers].

Acknowledgements

- Frank Ruskey and Joe Sawada
- Wendy Myrvold

Final Thoughts

Summary

- Introduced concept of fixed-density de Bruijn cycle.
- Cool-lex order is the Grand-Daddy of fixed-density de Bruijn cycles as lexicographic order is the Grand-Daddy of de Bruijn cycles.

Open Problems

- Improve efficiency of this construction for fixed-density binary strings.
- Is this construction the first in lexicographic / cool-lex order?
- Maximize/minimize the number of 1s in a fixed-density de Bruijn cycle.
- Maximize/minimize the shifts of length N in resulting order of $\mathbb{B}(N, D)$.
- Existence of *density-range de Bruijn Cycles* for $\bigcup_{i=\min}^{\max} \mathbb{B}(N, i)$?
- Generalize to shorthand universal cycles for multiset permutations.
  - Binary multiset case solved in this talk.
  - Set permutation case solved [Ruskey-W. 2009] [Bell-ringers].

Acknowledgements

- Frank Ruskey and Joe Sawada
- Wendy Myrvold

Final Thoughts

Summary

- Introduced concept of fixed-density de Bruijn cycle.
- Cool-lex order is the Grand-Daddy of fixed-density de Bruijn cycles as lexicographic order is the Grand-Daddy of de Bruijn cycles.

Open Problems

- Improve efficiency of this construction for fixed-density binary strings.
- Is this construction the first in lexicographic / cool-lex order?
- Maximize/minimize the number of 1s in a fixed-density de Bruijn cycle.
- Maximize/minimize the shifts of length N in resulting order of $\mathbb{B}(N, D)$.
- Existence of *density-range de Bruijn Cycles* for $\bigcup_{i=\min}^{\max} \mathbb{B}(N, i)$?
- Generalize to shorthand universal cycles for multiset permutations.
  - Binary multiset case solved in this talk.
  - Set permutation case solved [Ruskey-W. 2009] [Bell-ringers].

Acknowledgements

- Frank Ruskey and Joe Sawada
- Wendy Myrvold