

Hash Tables

Typically, the insertion and find algorithms will attempt to locate a valid location and terminate after m tries.

With linear probing this denotes the fact that every single one of the m locations in the table was examined:

- insertion: this signifies there is no open location to insert into
- find: this signifies no entry matches the target key

Quadratic Probing

Insert the following keys into the initially empty hash table below, with $m = 11$. Assume the hash function is $h(k) = k \bmod m$

43, 25, 14, 91, 07, 33, 58, 17, 03

--	--	--	--	--	--	--	--	--	--	--

0 1 2 3 4 5 6 7 8 9 10

--	--	--	--	--	--	--	--	--	--	--

0 1 2 3 4 5 6 7 8 9 10

--	--	--	--	--	--	--	--	--	--	--

0 1 2 3 4 5 6 7 8 9 10

--	--	--	--	--	--	--	--	--	--	--

0 1 2 3 4 5 6 7 8 9 10

--	--	--	--	--	--	--	--	--	--	--

0 1 2 3 4 5 6 7 8 9 10

Quadratic Probing Pros and Cons

Pros:

Cons:

Double Hashing

Insert the following keys into the hash table below, The hash function is $h(k) = k \bmod m$. The second hash function is $h_2(k) = 5 - (k \bmod 5)$

43, 25, 14, 91, 07, 33, 58, 17, 03

--	--	--	--	--	--	--	--	--	--	--

0 1 2 3 4 5 6 7 8 9 10

--	--	--	--	--	--	--	--	--	--	--

0 1 2 3 4 5 6 7 8 9 10

--	--	--	--	--	--	--	--	--	--	--

0 1 2 3 4 5 6 7 8 9 10

--	--	--	--	--	--	--	--	--	--	--

0 1 2 3 4 5 6 7 8 9 10

--	--	--	--	--	--	--	--	--	--	--

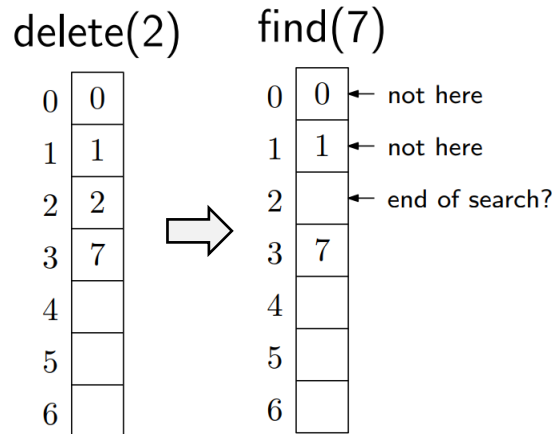
0 1 2 3 4 5 6 7 8 9 10

Pros:

Cons:

Deletion in Open Addressing

For this example, examine the table below, with the hash function $h(k) = k \bmod 7$, and that uses linear probing:



tombstone:

When searching, treat a tombstone as an occupied location.

Questions:

For insertion, can we overwrite the tombstone?

Will the hash table fill up with tombstones?