(1) (a) A transition diagram appears below for the TM specified by the 7-tuple $M = (\{q, q_0, q_1, q_2, q_{01}, q_{02}, q_{12}, q_{012}, r, f, z\}, \{0, 1\}, \{0, 1, B, X\}, \delta, q, B, \{f\})$. (Note that $q$ is used as the start state in this TM as opposed to $q_0$.)

(b) The TM proceeds in a number of passes starting from state $q$ with the tape head scanning the first non-blank symbol. The TM moves the tape head to the right during a pass until it replaces the first 0, first 1, and first 2 by an $X$. The states $q, q_0, q_1, q_2, q_{01}, q_{02}, q_{12}, q_{012}$ keep track of which symbols have been replaced by $X$ during the pass. Once successfully replacing these three symbols by $X$ the TM will be in state $q_{012}$ and there are two cases to consider.

- If the next symbol to the right is the blank, then the TM moves into state $z$. If the TM is in state $z$, then there must be no remaining copies of at least one of 0, 1, 2 on the tape. Thus, when the TM is in state $z$ it checks to see if all of the symbols on the tape are $X$, and if so it moves to state $f$ to accept.

- If the next symbol to the right is not the blank, then it must be one of 0, 1, 2. The TM reads this symbol and moves into state $r$ which proceeds to rewind the tape head to the beginning of the tape. Once the tape head is properly rewound, the TM moves into state $q$ to begin the next pass. The TM does not accept a string if a pass ends without finding occurrences of 0, 1, and 2 (unless the initial contents of the tape was blank).
(c) The following IDs show that 201210 is accepted.

\[ q_{201210} \vdash XXq_{021210} \vdash XXXq_{012210} \vdash XXrX210 \vdash XrXX210 \vdash rXXX210 \vdash rBXXX210 \vdash q XXXX210 \vdash XqXXX210 \vdash XXq210 \vdash XXXXq_{210} \vdash XXXXXzXX \vdash XXXzzXXX \vdash XXzXXX \vdash XzXXXXX \vdash z XXXXXX \vdash f XXXXXX \vdash \text{halt} \]

(d) The following IDs show that 10102 is not accepted.

\[ q_{10102} \vdash Xq_{10102} \vdash XXq_{10102} \vdash XX1q_{0102} \vdash XX10q_{012} \vdash XX10Xq_{012} B \vdash XX10rX \vdash XX1r0X \vdash \text{halt} \]

(2) (a) The binary representation of 11160850 is 10101001001100010010.

(b) The binary string \( w_{11160850} = 01010100100110100010010 \).

(c) The binary string \( w_{11160850} \) does not encode a valid Turing machine, so the transition diagram appears below.

(d) \( w_4 = 00 \notin \mathcal{L}(M_{11160850}) \) because \( \mathcal{L}(M_{11160850}) = \emptyset \).

Note: The intended question was to start with the decimal value 44643402.

(a) The binary representation of 44643402 is 10101010010011010001010.

(b) The binary string \( w_{44643402} = 01010100100110001001010 \).

(c) The binary string \( w_{44643402} \) does not encode a valid Turing machine, so the transition diagram appears below.

(d) \( w_4 = 00 \in \mathcal{L}(M_{44643402}) \) due to the following IDs

\[ q_1 00 \vdash 1 q_1 0 \vdash 11 q_1 B \vdash 1 q_2 10 \vdash \text{halt} \]

since \( q_2 \) is the final state.

(3) (a) We define the ID so that a new symbol \( \# \notin (\Gamma \cup Q) \) appears to the left of the symbol that the second tape head is scanning. If the two tape heads are scanning the same cell, then we choose to write \( \#q \) to the left of this cell where \( q \in Q \) is the current state. To formalize these conventions, suppose the area of interest on the tape is \( w_1 w_2 \ldots w_n \), the 2HTM is in state \( q \), the first head is scanning \( w_i \) and the second head is scanning \( w_j \). Now we consider the various cases.

If \( i = j \), then the ID is

\[ w_1 w_2 \ldots w_{i-1} \# q w_i w_{i+1} \ldots w_n. \]

If \( i < j \), then the ID is

\[ w_1 w_2 \ldots w_{i-1} q w_i w_{i+1} \ldots w_{j-1} \# w_j w_{j+1} \ldots w_n. \]
If \( j < i \), then the ID is
\[
w_1w_2 \ldots w_{i-1}\#w_iw_{i+1} \ldots w_{j-1}qw_jw_{j+1} \ldots w_n.
\]

(b) When defining \( \vdash \) we must take into account the special cases when \( i = n \) (the region of interest will grow), \( j = 1 \) and \( Z = B \) (the region of interest will shrink), and \( i = j \) (we choose to have the second tape head write overtop of what the first tape head writes). The various cases appear below.

If \( 1 < i < j < n \), then
\[
\begin{align*}
  &w_1w_2 \ldots w_{i-1}qw_iw_{i+1} \ldots w_{j-1}qw_jw_{j+1} \ldots w_n \\
  &w_1w_2 \ldots w_{i-1}Ypw_iw_{i+1}w_{i+2} \ldots w_{j-1}Z\#w_{j+1}w_{j+2} \ldots w_n.
\end{align*}
\]

If \( 1 < i = j < n \), then
\[
\begin{align*}
  &w_1w_2 \ldots w_{i-1}#w_iw_{i+1} \ldots w_n \\
  &w_1w_2 \ldots w_{i-1}Z\#pw_iw_{i+1}w_{i+2} \ldots w_n.
\end{align*}
\]

If \( 1 < j < i < n \), then
\[
\begin{align*}
  &w_1w_2 \ldots w_{i-1}#w_iw_{i+1} \ldots w_{j-1}qw_jw_{j+1}w_{j+2} \ldots w_n \\
  &w_1w_2 \ldots w_{i-1}Z\#w_{i+1}w_{i+2} \ldots w_{j-1}Ypw_{j+1}w_{j+2} \ldots w_n.
\end{align*}
\]

If \( 1 < j < i = n \), then
\[
\begin{align*}
  &w_1w_2 \ldots w_{i-1}#w_iw_{i+1} \ldots w_{n-1}qw_n \\
  &w_1w_2 \ldots w_{i-1}Z\#w_{i+1}w_{i+2} \ldots w_{n-1}ZpB.
\end{align*}
\]

If \( 1 = j < i < n \) and \( Z = B \), then
\[
\begin{align*}
  &#w_1w_2 \ldots w_{j-1}qw_jw_{j+1} \ldots w_n \\
  &#w_2 \ldots w_{j-1}Ypw_{j+1}w_{j+2} \ldots w_n.
\end{align*}
\]

If \( 1 = j < i = n \) and \( Z = B \), then
\[
\begin{align*}
  &#w_1w_2 \ldots w_{n-1}qw_n \\
  #w_2 \ldots w_{j-1}Ypw_{j+1}w_{j+2} \ldots w_n.
\end{align*}
\]

(c) Using our definition previous answers we can define the language of the 2HTM as \( \mathcal{L}(M) = \{ w \in \Sigma^* \mid #q_0w \vdash^* \alpha f \beta \text{ where } \alpha, \beta \in \{ \Gamma \cup \{ \# \} \}^* \text{ and } f \in F \} \). Notice that in our definition the initial ID is \#q_0w, which means we have chosen the two tape heads to start scanning the first cell in the input.

(d) The language of binary palindromes \( \mathcal{L} = \{ w \in \{0, 1\}^* \mid w = w^{\text{reverse}} \} \) could be decided more efficiently using a 2HTM since the 2HTM could avoid scanning back-and-forth. More specifically, if \( w \) is a binary string and \( |w| = n \), then a 2HTM could decide \( \mathcal{L} \) in at approximately \( 2n \) steps. This could be done by having the second head move to the end of the tape, and then comparing the symbols at position \( i \) and \( n - i + 1 \) on the tape for \( i = 1, 2, \ldots, \lfloor \frac{n}{2} \rfloor \). On the other hand, a regular TM would require approximately \( n^2 \) steps. This is because it would take \( n - i \) steps to compare the symbols at positions \( i \) and \( n - i + 1 \) for \( i = 1, 2, \ldots, \lfloor \frac{n}{2} \rfloor \). (Additional states could be used to reduce some of this work, but could not avoid the fact that the regular TM would have to scan back-and-forth.)
(4) Consider the following language where \( L_d \) is the diagonalization language
\[
L = \{x \in \{0, 1\}^* \mid x = 1w \text{ and } w \in L_d \} \cup \{x \in \{0, 1\}^* \mid x = 0w \text{ and } w \notin L_d \}.
\]

To understand \( L \) it is helpful to understand the role of the first bit in each string in the language. If the first bit is 1, then the remaining portion of the string must be in \( L_d \). On the other hand, if the first bit is 0, then the remaining portion of the string must not be in \( L_d \). Now we prove that \( L \) is not acceptable. For the sake of contradiction, suppose \( L \) is acceptable. Then there exists a Turing machine \( M \) such that \( L(M) = L \). Consider the Turing machine \( M_d \) that behaves as follows: Given input \( w \) it accepts if \( 1w \) is accepted by \( M \), and it does not accept if \( 1w \) is not accepted by \( M \).

It is clear that \( L(M_d) = L_d \). However, we know that \( L_d \) is not acceptable. Therefore, our initial assumption that \( L \) is acceptable must not be true.

The complement of \( L \) is
\[
\overline{L} = \{(x \in \{0, 1\}^* \mid x = 0w \text{ and } w \in L_d \} \cup \{x \in \{0, 1\}^* \mid x = 1w \text{ and } w \notin L_d \} \cup \{\varepsilon\}.
\]

Now we prove that \( \overline{L} \) is not acceptable. For the sake of contradiction, suppose \( \overline{L} \) is acceptable. Then there exists a Turing machine \( M \) such that \( L(M) = \overline{L} \). Consider the Turing machine \( M_d \) that behaves as follows: Given input \( w \) it accepts if \( 0w \) is accepted by \( M \), and it does not accept if \( 0w \) is not accepted by \( M \).

It is clear that \( L(M_d) = L_d \). However, we know that \( L_d \) is not acceptable. Therefore, our initial assumption that \( \overline{L} \) is acceptable must not be true.

(5) To illustrate the reductions used in the problem, we will refer to the following directed graph and use \( i = 1 \) and \( j = 3 \).

(a) If \( b \in \{0, 1\}^* \) can be expressed as \( b = g1110^i10^j \) where \( g \) encodes a directed graph, then let \( f(b) = 0^i10^j1b \). In other words, the function \( f \) simply adds an arc from \( v_j \) to \( v_i \). If \( b \) cannot be expressed in this way, then let \( f(b) = b \). For example, the result of applying \( f \) to the above example appears below. (In this example, there is no directed path from \( v_1 \) to \( v_3 \) in the original directed graph, so there is no directed cycle containing \( v_1 \) and \( v_3 \) in the directed graph below.)
Now we prove that \( b \in \mathcal{L}_p \iff f(b) \in \mathcal{L}_c \).

- If \( b \in \mathcal{L}_p \), then there is a directed path from \( v_i \) to \( v_j \) in the directed graph encoded by \( g \). This directed path together with the arc from \( v_j \) to \( v_i \) provides a directed cycle containing \( v_i \) and \( v_j \) in the directed graph encoded by \( 0^{10}1^{11}g \), and therefore \( f(b) = 0^{10}1^{11}b \in \mathcal{L}_c \).

- If \( b \not\in \mathcal{L}_p \), then there is no directed path from \( v_i \) to \( v_j \) in the directed graph encoded by \( g \). Therefore, there is also no directed path from \( v_i \) to \( v_j \) in the directed graph encoded by \( 0^{10}1^{11}g \). Therefore, there is no directed cycle containing \( v_i \) and \( v_j \) in the directed graph encoded by \( 0^{10}1^{11}g \), and so \( f(b) = 0^{10}1^{11}b \not\in \mathcal{L}_c \).

(b) If \( b \in \{0, 1\}^* \) can be expressed as \( b = g1110^{i}10^{j} \) where \( g \) encodes a directed graph \( G \) with \( n \) vertices, then let \( f(b) = g'1110^{i+n} \), where \( g' \) encodes a directed graph \( G' \) defined as follows. Let \( G' \) be obtained by making a second copy of \( G \) and relabelling each vertex \( v_x \) by \( v_{x+n} \) in the second copy. Also, add an arc from \( v_j \) to its duplicate \( v_{j+n} \) in the second copy. If \( b \) cannot be expressed in this way, then let \( f(b) = b \). For example, the result of applying \( f \) to the above example appears below. (In this example, there is no directed cycle containing \( v_1 \) and \( v_3 \) in the original directed graph, so there is no directed path from \( v_1 \) to \( v_5 \) in the directed graph below.)

Now we prove that \( b \in \mathcal{L}_c \iff f(b) \in \mathcal{L}_p \).

- If \( b \in \mathcal{L}_c \), then there is a directed cycle including \( v_i \) and \( v_j \) in the directed graph \( G \). This directed cycle includes a directed path from \( v_i \) to \( v_j \) together with a directed path from \( v_j \) to \( v_i \). Therefore, there is a directed path from \( v_i \) to \( v_j \) in \( G' \), and also a directed path from \( v_{j+n} \) to \( v_{i+n} \) in \( G' \). The first of these directed paths, followed by the arc from \( v_j \) to \( v_{j+n} \), followed by the second of these directed paths collectively provide a directed path from \( v_i \) to \( v_{i+n} \) in \( G' \). Therefore, \( f(b) \in \mathcal{L}_p \).

- If \( b \not\in \mathcal{L}_c \), then there is no directed cycle including \( v_i \) and \( v_j \) in the directed graph \( G \). Therefore, there is either no directed path from \( v_i \) to \( v_j \) in \( G \), or there is no directed path from \( v_j \) to \( v_i \) in \( G \).
  - If there is no directed path from \( v_i \) to \( v_j \) in \( G \), then there is no directed path from \( v_i \) to \( v_j \) in \( G' \). Furthermore, since the only arc between the
two copies of $G$ in $G'$ is from $v_j$ to $v_{j+1}$, then there is no directed path from $v_i$ to $v_{i+n}$ in $G'$. Therefore, $f(b) \notin L_c$.

- If there is no directed path from $v_j$ to $v_i$ in $G$, then there is no directed path from $v_{j+n}$ to $v_{i+n}$ in $G'$. Furthermore, since the only arc between the two copies of $G$ in $G'$ is from $v_j$ to $v_{j+1}$, then there is no directed path from $v_i$ to $v_{i+n}$ in $G'$. Therefore, $f(b) \notin L_c$. 