7.1
a. True
b. False
c. False
d. True
e. True
f. True

7.2
a. False
b. True
c. False
d. True
e. False
f. True

7.6 Assume \( L_1 \) and \( L_2 \) are in \( P \) and let \( M_1 \) and \( M_2 \) be single-tape deterministic TM’s that decide \( L_1 \) and \( L_2 \), respectively. Assume \( M_1 \) runs in \( O(n^j) \) time and \( M_2 \) runs in \( O(n^k) \) time.

a. \( P \) is closed under union. Let \( M_3 \) be the following TM
   \[
   M_3 = \text{“On input } w \text{”}
   \]
   1. Run \( M_1 \) on input \( w \)
   2. Run \( M_2 \) on input \( w \)
   3. If either returned accept, then accept
   4. Otherwise, reject

   Then \( M_3 \) is a deterministic single-tape TM that decides \( L_1 \cup L_2 \) in \( O(n^j) + O(n^k) = O(n^{\max\{j,k\}}) \) time. Therefore, \( L_1 \cup L_2 \) is in \( P \).

b. \( P \) is closed under concatenation. Let \( M_4 \) be the following TM
   \[
   M_4 = \text{“On input } w \text{”}
   \]
   1. For \( k = 0 \) to \( |w| \)
   2. Divide \( w \) into \( u \) and \( v \) where \( |u| = k \) and \( |v| = |w| - k \)
   3. Run \( M_1 \) on input \( u \)
   4. Run \( M_2 \) on input \( v \)
   5. If both returned accept, then accept
   6. Next \( k \)
   7. If no deconstruction of \( w \) was accepted, reject.

   Then \( M_4 \) is a deterministic single-tape TM that decides \( L_1 \circ L_2 \) in \( |w|(O(n^j) + O(n^k)) = O(n^{\max\{j,k\}}) = O(n^{\max\{j,k\}+1}) \) time. Therefore, \( L_1 \circ L_2 \) is in \( P \).
c. \(P\) is closed under complement. Let \(M_5\) be the following TM
\[M_5 = \text{“On input } w\]
1. Run \(M_1\) on input \(w\)
3. If \(M_1\) returned accept, then reject
4. Otherwise, accept.”

Then \(M_5\) is a deterministic single-tape TM that decides \(\neg L_1\) in \(O(n^1)\) time. Therefore, \(\neg L_1\) is in \(P\).

7.9 Step 1 can be done in \(O(1)\) time. Step 3 takes \(O(|V|)\) time, where \(|V|\) is the number of nodes in \(G\). Since step 3 is repeated until no new nodes are marked, it is repeated at most \((|V|-1)\) times. Step 4 takes \(O(|V|)\) time to complete. Therefore, the entire algorithm takes \(O(1) + (|V|-1) \times O(|V|) + O(|V|) = O(|V|^2)\) time to complete. Therefore, CONNECTED is in \(P\).

7.12 The following algorithm determines if \(a^b \equiv c \pmod{p}\).
\[M = \text{“on input } <a,b,c,p>\]
1. Let \(j = \text{length}(b)\) \(\% j = \lceil \log_2 b \rceil + 1\)
2. Let \(m = a \pmod{p}\)
3. If \(p\) is even, let \(r = 1\)
4. If \(p\) is odd, let \(r = m\)
5. For \(i = 1\) to \(j\)
6. Let \(m = m^2 \pmod{p}\)
7. If the \(i^{th}\) byte of \(b\) (reading right to left) is a 1
8. \(r = r \times m \pmod{p}\)
9. Next
10. Let \(s = c \pmod{p}\)
11. If \(r = s\), then accept
12. Else, reject.”

This algorithm accepts the input iff \(a^b \equiv c \pmod{p}\). Each step can be done in polynomial time. The loop is executed \(\text{length}(b)\) times and is therefore polynomial in the length of the problem. Note that if I had looped from 1 to \(b\) and multiplied by \(a\) in each loop, the number of iterations would have been exponential in the length of the problem!