Question 1: Exercise 5.13

This question is tricky. So, I will give you some direction.

\[ T = \{ \langle M \rangle | M \text{ is a TM that has a state that is never entered} \} \]

Assume that \( T \) is decidable and that \( R \) decides it.

We will show that we can use \( R \) to decide \( A_{TM} \).

Give a construction of TM \( S \) that will decide \( A_{TM} \) using \( R \) as a subroutine.

This is a difficult question.

Question 2: Problem 5.23

Question 3: Exercise 7.6

Just do concatenation.

Give a complete proof starting with:

Let \( L_1 \) and \( L_2 \) be in \( P \).

So, there exists deterministic Turing machines \( M_1 \) and \( M_2 \) that decide \( L_1 \) and \( L_2 \) respectively.

And \( M_1 \) and \( M_2 \) run in polynomial time.

Set \( k \) so that \( M_1 \) and \( M_2 \) are in \( \text{TIME}(n^k) \).

In your proof, give a high-level description of the TM, and argue why it is polynomial time.

Question 4: NP and Complementation

Part a:

The following is an incorrect proof that NP is closed under complementation. What is wrong with this proof?

Let \( L_1 \in \text{NP} \).

Let \( N_1 \) be a polynomial time nondeterministic Turing machine that decides \( L_1 \).

Set \( k \) so that \( N_1 \) is in \( \text{NTIME}(n^k) \).

Construct \( N \) that decides \( \overline{L_1} \) as follows:

\[ N = \text{“On input } w \text{”} \]

1. Run \( N_1 \) on \( w \)
2. If \( N_1 \) rejects, accept; else reject

\( N_1 \) is a decider and so will always halt, and will do so in \( O(n^k) \). So, \( N \) will also take time \( O(n^k) \).

Part b:

If you have a verifier \( V_1 \) for \( L_1 \in NP \), why can’t you use this verifier to construct a verifier \( V \) for \( \overline{L_1} \)? Do not give a proof, just a rationale.
Question 5: Exercise 7.7: Verifier

Just do concatenation. Do this by constructing a verifier. Describe what the certificate should be, and give a high-level implementation of the verifier. Also, argue why your verifier is correct.

Question 6: Exercise 7.7: NTM

Just do concatenation. Let $L_1$ and $L_2$ be in NP. Show that $L = L_1 \circ L_2$ is in NP. For this question, construct a non-deterministic Turing machine $N$ that accepts the $L$. Unlike the solution to 7.15, do not make use of verifiers for $L_1$ and $L_2$, but construct $N$ using nondeterministic TMs $N_1$ and $N_2$ that decide $L_1$ and $L_2$ respectively.

To construct $N$ properly, make sure you do not make the same mistake as in the question titled ‘NP and Complementation’.

Question 7: Exercise 7.5

Question 8: Exercise 7.18

This is question 7.18 in the third edition, and 7.17 in the second edition.

Show that if $P = NP$, then every language $A \in P$, except $A = \emptyset$ and $A = \Sigma^*$, is NP-complete.

Question 9: Problem 7.21

This is question 7.20 in the second edition, and 7.21 in the third edition.

*Simple path* is defined on page 11.

For part a, give an algorithm that can decide it in polynomial time. Make sure you analyze the time complexity and say what it is.

For part b, you can use Theorem 7.55 that UHAMPATH is NP-Complete.

Question 10: Problem 7.41

This is question 7.41 in the third edition, and 7.39 in the second edition.

Question 11: Problem 7.38

This is question 7.38 in the third edition, and 7.36 in the second edition.