Homework #9

Raja Kantheti

Problem 1: P, NP, and NP-Complete

Assumptions:

- **Prob1** is in P.
- **Prob2** is in NP.
- **Prob3** is NP-Complete.

(a) There exists an algorithm for Prob2 with worst-case polynomial-time complexity.

- **If** $P = NP$**:** Yes. If $P = NP$, all problems in NP can be solved in polynomial time, including Prob2.
- **If** $P \neq NP$**:** Possibly yes, but it depends on the problem. Prob2 is in NP, so it might still be solvable in polynomial time even if $P \neq NP$.

(b) There exists an algorithm for Prob3 with worst-case polynomial-time complexity.

- **If** $P = NP$**:** Yes. Since Prob3 is NP-Complete, if $P = NP$, then all problems in NP (including NP-Complete problems like Prob3) can be solved in polynomial time.
- **If** $P \neq NP$**:** No. NP-Complete problems cannot be solved in polynomial time under the assumption that $P \neq NP$.

(c) There exists no polynomial-time reduction from Prob3 to Prob2.

- **If** $P = NP$**:** No. If $P = NP$, any NP problem (including NP-Complete problems like Prob3) can be reduced to another NP problem (such as Prob2) in polynomial time.
- **If** $P \neq NP$**:** No. Regardless of whether $P = NP$, there always exists a polynomial-time reduction from one NP-Complete problem (like Prob3) to another NP problem.

(d) Prob3 is NP-Hard.

- **If $P = NP$:** Yes. NP-Complete problems are, by definition, NP-Hard, and this property is independent of whether $P = NP$.
- **If $P \neq NP$:** Yes. The definition of NP-Completeness implies NP-Hardness, which remains true even if $P \neq NP$.

Problem 2: Prove that the Decision Version of the Traveling Salesperson Problem (TSP) is NP-Complete

Proving that the Decision Version of TSP is NP-Complete

To prove that the decision version of the Traveling Salesperson Problem (TSP) is NP-Complete, we must follow these steps:

1. Show that TSP is in NP.
2. Reduce a known NP-Complete problem to TSP in polynomial time.

Step 1: TSP is in NP

The decision version of TSP is defined as follows:

> **Given:** A set of n cities, a function $c(i, j)$ representing the distance between city i and city j for all pairs i, j, and a threshold k.
> **Question:** Is there a route that visits every city exactly once, returning to the starting city, with a total distance less than or equal to k?

To show that TSP is in NP, we verify that any "yes" instance of TSP can be checked in polynomial time. Let the candidate solution be a specific ordering of the cities $S = \{s_1, s_2, \ldots, s_n, s_1\}$. Verification involves:

- **Step 1: Validate the structure of the route:**
  - Ensure that all n cities are visited exactly once. This involves:
    * Checking that the sequence S contains $n + 1$ cities, where the last city equals the first.
    * Using a set or frequency counter to verify no duplicates or omissions among the n cities.
  - This step has a time complexity of $O(n)$.
- **Step 2: Compute the total distance:**
  - Use the distance function $c(i, j)$ to compute the total distance:

$$\text{Total distance} = \sum_{i=1}^{n} c(s_i, s_{i+1}),$$

    where $s_{n+1} = s_1$ to complete the cycle.

- This step requires n calls to $c(i, j)$, each taking $O(1)$ time, resulting in a total complexity of $O(n)$.

- **Step 3: Compare the total distance to the threshold k:**
  - A simple comparison operation takes $O(1)$ time.

Overall Complexity: The total verification time is:

$$T_{\text{verify}}(n) = O(n) + O(n) + O(1) = O(n).$$

Since verification is polynomial, TSP belongs to NP.

Step 2: Reduction from HAM-CYCLE to TSP

The Hamiltonian Cycle Problem (HAM-CYCLE) is a known NP-Complete problem. It asks:

> **Given:** An undirected, unweighted graph $G = (V, E)$.
> **Question:** Does there exist a cycle that visits every vertex exactly once?

To prove TSP is NP-Complete, we reduce HAM-CYCLE to TSP in polynomial time.

Reduction Construction:

Given a graph $G = (V, E)$:

1. Create a complete graph $G' = (V, E')$, where:
   - E' contains all possible edges between vertices in V.
   - For each edge $e \in E$ in the original graph G, assign a weight of 1.
   - For each edge $e \notin E$, assign a very large weight M, where $M > |V| \cdot$ max edge weight in G.
2. Set the threshold $k = |V|$, representing the total distance of the Hamiltonian cycle.

Reduction Correctness:

- If G has a Hamiltonian cycle, this corresponds to a route in G' where all edges have weight 1. The total distance of this route is exactly $|V|$, which is less than or equal to the threshold k.
- If G does not have a Hamiltonian cycle, any route in G' must use at least one edge with weight M. Since $M > k$, such a route will exceed the threshold.

Reduction Complexity:

- Constructing the complete graph G' requires adding $O(|V|^2)$ edges.
- Assigning weights to edges is also $O(|V|^2)$.
- Thus, the reduction runs in $O(|V|^2)$, which is polynomial.

Conclusion

1. TSP is in NP because a "yes" instance can be verified in $O(n)$.
2. HAM-CYCLE reduces to TSP in $O(|V|^2)$, which is polynomial.

By definition, TSP is NP-Complete.