1. Traveling salesperson problem (TSP)

There are n cities, and there is a positive (integral) cost c_{ij} of traveling from the i-th city to the j-th city. The costs need not be symmetric, that is, we may have $c_{ij} \neq c_{ji}$. Moreover, we may have $c_{ij} + c_{jk} \neq c_{ik}$. A salesperson starts at a given city, visits every city once and only once, and eventually comes back to the city from where the tour started. The objective of the salesperson is to minimize the total cost of traveling, that is, to find a minimum-cost Hamiltonian cycle in a complete weighted directed graph G on n vertices. This is an optimization problem. We consider the equivalent decision problem TSP that, given G and a positive integer k, decides whether G contains a (directed) Hamiltonian cycle of total cost $\leq k$. Prove that TSP is NP-complete.

2. Longest path problem

Let $G = (V, E)$ be a weighted graph (directed or undirected), s and t two vertices in V, and k a positive integer. Assume that the edge weights are positive (or non-negative). Prove that the problem of deciding whether G contains an s,t path of weight $\geq k$ is NP-complete.

3. Shortest path problem

Let $G = (V, E)$ be a weighted graph (directed or undirected), s and t two vertices in V, and k a positive integer. Consider the problem of deciding whether G contains an s, t path of weight $\leq k$. Justify whether this problem is NP-complete if

(a) all edge weights are positive (or non-negative),
(b) the edge weights may be positive, negative, or zero.

4. Let G be an undirected graph on n vertices. Which of the following problems is/are NP-complete? Justify.

(a) Decide whether G contains a clique of size $\geq n - 5$.
(b) Decide whether G contains an independent set of size $\geq n - 5$.
(c) Decide whether G contains a vertex cover of size $\geq n - 5$.
(d) Decide whether G contains a cycle of length $\geq n - 5$.
(e) Decide whether G contains a path of length $\geq n - 5$.

5. Partition problem

You are given n positive integers $a_1, a_2, \ldots, a_n$ such that

$$a_1 + a_2 + \cdots + a_n = A \text{ is even.}$$

Decide whether the given integers can be partitioned into two subcollections such that the sum of the integers in each subcollection is $A / 2$. Prove that PARTITION is NP-complete.

6. Bin-packing problem

You are given n objects of weights $w_1, w_2, \ldots, w_n$, and an infinite supply of bins each with weight capacity C. You want to pack all the objects in m bins such that m is as small as possible. Assume that each $w_i \leq C$.

(a) Frame an equivalent decision problem, and prove that the decision problem can be solved in polynomial time if and only if the optimization problem can be solved in polynomial time.

(b) Prove that the decision version is NP-complete.

7. Knapsack problem

A thief finds n objects of weights $w_1, w_2, \ldots, w_n$ and positive integer-valued profits $p_1, p_2, \ldots, p_n$. The thief has a knapsack of capacity C. The goal of the thief is to pack objects in the knapsack without exceeding its capacity, so as to maximize the profit of packed objects.

0,1 variant: Each object can be either packed or discarded.
Fractional variant: Any fraction of any object can be packed.

(a) Prove that the fractional knapsack problem is in P.
(b) Formulate an equivalent decision version of the 0,1 knapsack problem.
(c) Prove that the decision problem of Part (b) is NP-complete.

8. MAX-CUT problem

Let $G = (V, E)$ be an undirected graph. We want to find a cut X, Y of V such that the number of edges connecting X and Y is as large as possible.

(a) Propose a decision version of the MAX-CUT problem. Prove that the maximization problem can be solved in polynomial time if and only if the decision problem can be solved in polynomial time.

(b) Prove that the decisional MAX-CUT problem is NP-complete.

9. MAX-3-CUT problem

Let $G = (V, E)$ be an undirected graph, and k a positive integer. Decide whether V can be partitioned into three parts X, Y, Z such that the number of edges in E connecting vertices from different parts is $\geqslant k$. Prove that MAX-3-CUT is NP-complete.

10. Let $G = (V, E)$ be an undirected graph, and k a positive integer. You want to determine whether the removal of some k or fewer edges from E makes G bipartite. Prove that this problem is NP-complete.

11. Weighted MAX-CUT problem

Let $G = (V, E)$ be an undirected graph with each edge e carrying a positive weight w_e (assume to be a positive integer). The weight of a cut X, Y of V is the sum of the weights of the edges connecting X and Y. Let k be a positive integer. We want to decide whether G has a cut of weight $\geqslant k$. Prove that the weighted MAX-CUT problem is NP-complete.

12. Ferry-loading problem

Several vehicles wait in a long queue near a river bank. The lengths of the vehicles are l_1, l_2, l_3, …, l_n (positive integers) in that order from the beginning to the end of the queue. A big boat with two decks (left and right) comes to carry the vehicles across the river. Each deck of the boat has length L (again a positive integer). The vehicles must be loaded to the boat in the order they appear in the queue. For each vehicle, a decision is to be made about which deck it will join (provided that there is enough length remaining in that deck). The objective is to maximize the number of vehicles that can be loaded. As an example, take $L = 10$, and the first four vehicles having lengths 5, 5, 6, 4. Then all the four of them can be loaded ($5 + 5 = 6 + 4 = 10$). However, if the first two vehicles are loaded to different decks, then no other vehicle can be loaded. Let us consider the decision problem whether all of the vehicles can be loaded. Clearly, if

$$l_1 + l_2 + \cdots + l_n > 2L,$$

the answer is *No*. So assume that $l_1 + l_2 + \cdots + l_n \leq 2L$.

(a) Prove that the optimization problem is polynomial-time equivalent to the decision problem.

(b) Prove that the decision problem is NP-Complete.

13. Let C be a set of classes. Each class is specified by its duration (a positive integer), can be scheduled to start at any working time of the day, and must finish uninterrupted before the working time ends. Given a positive integer k, we want to find out whether k classrooms suffice for scheduling all the classes in C on a single day. Assume that the working time of a day is from 8:00am to 8:00pm.

Prove/Disprove: This problem is NP-Complete.

14. Let C again be a set of classes. Each class is specified by a start time and an end time (both in the range from 8:00am to 8:00pm), and must be scheduled during this specified interval. Given a positive integer k, we want to find out whether k classrooms suffice for scheduling all the classes in C on a single day.

Prove/Disprove: This problem is NP-Complete.

15. DNFSAT

A Boolean formula φ is given in the disjunctive normal (sum-of-products) form. Prove/Disprove: The problem of deciding whether φ is satisfiable is NP-complete.