

1. You work in the transportation department and are asked by your boss to design a railway query system for tourists. Your country's railway network is **very huge** and so it is often difficult for a tourist to find his/her way around. Your system is supposed to help them out. Specifically, when a tourist asks about how to go from station s_1 to station s_k , your system should output a list of stations s_1, s_2, \dots, s_k so that s_i and s_{i+1} are connected by a railway and there is no other station between them on that railway. For system efficiency, your boss decides that if the system has found out how to go from station s_1 to station s_k for one tourist, then your system should store this solution reported. Subsequently, when another tourist comes along and ask the same question again, your system can then retrieve this solution from the storage without solving the same problem over again. **Since this is a pilot system, you do not need to worry about the solution quality, i.e., it is acceptable as long as it is feasible.**

Describe how to model the above problem as a graph problem. Describe how to answer a tourist's query when the same query has not been asked before. Describe how to organize a data structure for storing solutions to queries so that queries asked before can be recognized **efficiently** and then the relevant solution can be retrieved **efficiently**. **The storage overhead must be kept at minimum.**

2. Let $G = (V, E)$ be an undirected graph. A *node cover* of G is a subset W of the vertex set V such that every edge in E is incident to some vertex in W . A *minimum node cover* is a node cover with the fewest number of vertices.
 - (a) Given a subset $U \subset V$ that is not a node cover. Give a greedy strategy to select a vertex $w \in V \setminus U$ such that $U \cup \{w\}$ is as close to a node cover as possible.
 - (b) Extend your strategy into a greedy algorithm to find a node cover for G .
 - (c) Show that your greedy algorithm does not always find the minimum node cover.
3. (Due 15 October 1998) Suppose that we are given a cable network of n sites connected by duplex communication channels. Unfortunately, the communication channels are not perfect. Each channel may fail with certain probability (also given in the input). The probabilities of failure may differ for different channels and they are mutually independent events. One of the n sites is the central station and your problem is to compute the most reliable paths from the central station to all other sites. Design an algorithm for solving this problem and analyze its time and space complexities.