CS514: Design and Analysis of Algorithms

Arijit Mondal

Dept of CSE

arijit@iitp.ac.in

https://www.iitp.ac.in/~arijit/

Problem-1

- We have three containers whose sizes are 10 liters, 7 liters, and 4 liters, respectively. The 7-liter and 4-liter containers start out full of water, but the 10-liter container is initially empty. We are allowed one type of operation: pouring the contents of one container into another, stopping only when the source container is empty or the destination container is full. We want to know if there is a sequence of pourings that leaves exactly 2 liters in the 7- or 4-liter container.
- Model this as a graph problem: give a precise definition of the graph involved and state the specific question about this graph that needs to be answered. What algorithm should be applied to solve the problem?

Problem-2

- A factory must periodically replace its equipment because of machine wear. As a machine ages, it breaks down more frequently and so becomes more expensive to operate. Furthermore, with years its salvage value decreases.
- To find a plan that minimizes the total cost of buying, selling, and operating the machine over a planning horizon of n years assuming that the factory must have one machine in service at all times. Selling / procurement of machine can happen at the end of a year.

Problem-3

- Bus driver allocation problem: The following table illustrates a number of possible duties for the drivers of a bus company. We wish to ensure at the lowest possible cost, that at least one driver is on duty for each hour of the planning period (0900 to 1700)

Hours	9-13	9-11	12-15	12-17	14-17	13-16	16-17
Cost	300	180	210	380	200	340	90

Problem-4

- A farmer wishes to transport a truckload of eggs from one city to another city through a given road network. The truck will incur a certain amount of breakage on each road segment. Let $w_{j,k}$ denote the fraction of the eggs broken if the truck traverses the road segment (j, k)
- How should the truck be routed to minimize the total breakage?

Problem-5

- A bipartite graph is a graph $G = (V, E)$ whose vertices can be partitioned into two sets $(V = V_1 \cup V_2 \text{and} V_1 \cap V_2 = \emptyset)$ such that there are no edges between vertices in the same set (for instance, if $u, v \in V_1$, then there is no edge between u and v).
- Give a linear-time algorithm to determine whether an undirected graph is bipartite.

Problem-6

- For each node u in an undirected graph, let `twodegree[u]` be the sum of the degrees of u's neighbors. Show how to compute the entire array of `twodegree[]` values in linear time, given a graph in adjacency list format.

Problem-7

- Suppose a CS curriculum consists of n courses, all of them mandatory. The prerequisite graph G has a node for each course, and an edge from course v to course w if and only if v is a prerequisite for w. Find an algorithm that works directly with this graph representation, and computes the minimum number of semesters necessary to complete the curriculum (assume that a student can take any number of courses in one semester). The running time of your algorithm should be linear.

Problem-8

- Give an efficient algorithm that takes as input a directed acyclic graph $G = (V, E)$, and two vertices $s, t \in V$, and outputs the number of different directed paths from s to t in G.

Problem-9

- Here's a proposal for how to find the length of the shortest cycle in an undirected graph with unit edge lengths. When a back edge, say (v, w), is encountered during a depth-first search, it forms a cycle with the tree edges from w to v. The length of the cycle is level[v] $-$ level[w] $+$ 1, where the level of a vertex is its distance in the DFS tree from the root vertex. This suggests the following algorithm:
  - 1. Do a depth-first search, keeping track of the level of each vertex.
  - 2. Each time a back edge is encountered, compute the cycle length and save it if it is smaller than the shortest one previously seen.
- Comment on this algorithm