Design and Analysis of Algorithms 2020
Problem Set 4

Chennai Mathematical Institute

October 24, 2020

Problem 1. The degree of a node is the number of children it has. Show that in any binary tree the number of leaves is one more than the number of nodes of degree two.

Problem 2. Adam and Eve are taking a roadtrip from Kolkata to Chennai. Because it's a 33-hour drive, Adam and Eve decide to switch off driving at each rest stop they visit; however, because Adam has a better sense of direction than Eve, he should be driving both when they depart and when they arrive (to navigate the city streets). Given a route map represented as a weighted undirected graph $G = (V, E, w)$ with positive edge weights, where vertices represent rest stops and edges represent routes between rest stops, devise an efficient algorithm to find a route (if possible) of minimum distance between Kolkata and Chennai such that Adam and Eve alternate edges and Adam drives the first and last edge.

Problem 3. Suppose you are given a connected graph G, with edge costs that you may assume are all distinct. G has n vertices and m edges. A particular edge e of G is specified. Give an algorithm with running time $\mathcal{O}(m+n)$ to decide whether e is contained in a minimum spanning tree of G.

Problem 4. Suppose you are given a connected graph G, with edge costs that are all distinct. Prove that G has a unique minimum spanning tree.

Problem 5. Consider the Minimum Spanning Tree Problem on an undirected graph $G = (V, E)$, with a cost $c_e \geqslant 0$ on each edge, where the costs may not all be different. If the costs are not all distinct, there can in general be many distinct minimum-cost solutions. Suppose we are given a spanning tree $T \subseteq E$ with the guarantee that for every $e \in T$, e belongs to some minimum-cost spanning tree in G. Can we conclude that T itself must be a minimum-cost spanning tree in G? Give a proof or a counterexample with explanation.

Problem 6. You are given a vertex-weighted graph. Consider the following definitions.

- Independent set: a set of vertices in a graph, no two of which are adjacent.
- Weight of independent set: sum of weights of vertices in the set.
- Max-weight independent set problem: Find an independent set which has maximum weight.

Consider the following greedy algorithm for max-weight independent set problem:

1. Start with an empty set X.
2. For each vertex v, in decreasing order of weight: add vertex v to the set X if v is not adjacent to any vertex in X.
3. Return X, $weight(X)$.

Show that the above greedy approach for max-weight independent set problem is not optimal, by exhibiting a small counterexample. How badly suboptimal can greed be, relative to optimal?

Problem 7. Mr. Greedy Barber has to schedule n clients for haircuts for the next day. These n clients are regular customers, so for each customer i, Mr. Barber know the exact time h_i it would take him to cut i's hair. Mr. Barber is also gunning for a record, so he has decided not to waste any time moving from one haircut to the next. Help Mr. Barber design an $\mathcal{O}(n \log n)$ time algorithm that schedules all the n clients, which provably minimizes the sum of the service time for every client, where the service time for client i is h_i plus the time i had to wait before i's turn came by. You should assume that all the clients came into the shop at the same time. As usual, you must prove the optimality of your algorithm. Here is a simple example: say $n = 3$ and

$$h_1 = 5, \quad h_2 = 10, \quad h_3 = 4.$$

Now consider the schedule

$$1, 2, 3$$

i.e. 1 get the haircut first and then leaves, 2 gets his haircut after 1 and 3 gets his hair cut after 2. Note that the service time for 1 is 5. The service time for client 2 is his wait time (which is 5) and h_2, which is 15. The service time for client 3 is his wait time (which is $5 + 10 = 15$) plus h_3, which is 19. Thus, the sum of the service times for this schedule is $5 + 15 + 19 = 39$. This schedule, however, is not optimal.

Problem 8. Given a directed graph $G = (V, E)$, a vertex $s \in V$ is called a sink if there are incoming edges from every other vertex to s but no outgoing edge from s.

- As a warmup present an $\mathcal{O}(n^2)$ algorithm to find out if G has a sink and if so, to output it. (Recall that $n = |V|$).

- Now present an $\mathcal{O}(n)$ time algorithm for the same problem.

Problem 9. Chefina has two sequences $A_1, A_2, \ldots, A_N$ and $B_1, B_2, \ldots, B_N$. She views two sequences with length N as identical if, after they are sorted in non-decreasing order, the i-th element of one sequence is equal to the i-th element of the other sequence for each i.

To impress Chefina, Chef wants to make the sequences identical. He may perform the following operation zero or more times: choose two integers i and j ($1 \leqslant i, j \leqslant N$) and swap A_i with B_j. The cost of each such operation is $\min(A_i, B_j)$.

You have to find the minimum total cost with which Chef can make the two sequences identical. You might get more credit for better complexity.

Problem 10-15. Exercises $15.3, 15.4, 15.5, 16.1, 16.3, 16.4$ from CLRS.

Problem 16-19. Exercises $5.27, 5.31, 6.12, 6.21$ from DPV.

Problem 20-21. Exercises $6.1, 6.3$ from KD.