COL351: Analysis and Design of Algorithms **Fall 2024**

Tutorial Sheet 7

Announced on: Sept 07 (Sat)

Problems marked with ($\star$) will not be asked in the tutorial quiz.

1. This problem describes the *cut property* of minimum spanning trees (MSTs).

   Let $G = (V, E)$ be a connected and weighted graph where the edge costs may not be distinct. For any subset $S \subseteq V$ of the vertices, let $\bar{S} := V \setminus S$ denote the vertices not in S. Let $E(S, \bar{S})$ denote the set of all edges with one endpoint each in S and $\bar{S}$. Show that if there exists a unique edge $e \in E(S, \bar{S})$ with the smallest weight among the edges in $E(S, \bar{S})$, then e must belong to every MST of graph G.

2. Consider the following alleged proof of the cut property from Problem 1.

   Let T be a minimum spanning tree that does not contain e. Since T is a spanning tree, it must contain an edge f with one end in S and the other end in $\bar{S}$. Since e is the cheapest edge with this property, we have $c_e < c_f$, and hence $T \setminus \{f\} \cup \{e\}$ is a spanning tree that is cheaper than T.

   Is this proof correct? If not, then explain the error.

3. Suppose you are given a connected weighted graph $G = (V, E)$ with a distinguished vertex s where all edge costs are positive and distinct. Is it possible for a tree of shortest paths from s and a minimum spanning tree in G to not share any edges? If so, give an example. If not, give a reason.

4. Consider a connected undirected graph G with distinct edge costs. Design a linear-time algorithm to find a *minimum bottleneck spanning tree*, which is a spanning tree T that minimizes the maximum edge cost $\max_{e \in T} c_e$. You can assume access to a subroutine `FindMedian` that, given as input an unsorted list of n numbers, computes the median in $\mathcal{O}(n)$ time.

   For the tutorial quiz, you can provide a plain English description of the algorithm as your solution. You don't need to write a complete pseudocode, but you are welcome to include it in addition to or instead of the English description. Additionally, please provide a brief justification (one or two sentences) for the correctness and running time of the algorithm.

5. Suppose we are given the minimum spanning tree T of a given graph $G = (V, E)$ with distinct edge costs, and a new edge $e = (u, v)$ of cost c that we will add to G. Design an $\mathcal{O}(|V|)$ algorithm to find the minimum spanning tree of the graph $G + e$. Clearly write the

pseudocode and briefly justify (in one or two sentences) why the algorithm is correct and has the desired running time.

6. You wish to drive from point A to point B along a highway minimizing the time that you are stopped for gas. You are told beforehand the capacity C of your gas tank in liters, your rate F of fuel consumption in liters/kilometer, the rate r in liters/minute at which you can fill your tank at a gas station, and the locations $A = x_1, x_2, \ldots, x_{n-1}, x_n = B$ of the gas stations along the highway. So, if you stop to fill your tank from 2 liters to 8 liters, you would have to stop for $6/r$ minutes. Consider the following two algorithms:

   a) Stop at every gas station, and fill the tank with just enough gas to make it to the next gas station.

   b) Stop if and only if you don't have enough gas to make it to the next gas station and if you stop fill the tank up all the way.

   For each algorithm, either prove or disprove that the algorithm correctly solves the problem. Your proof of correctness must use an exchange argument.

7. ($\star$) In tutorial sheet 6, you saw an example where the "top down" divide-and-conquer strategy for the optimal prefix-free code problem is suboptimal. You have also seen in class that the Huffman coding algorithm is optimal.

   However, you love divide-and-conquer algorithms and can't stop thinking about them. So instead, you devise the following proposal: Define a *median* symbol whose frequency, along with the frequencies of all other symbols with higher (respectively, lower) frequency, adds up to at least 50% of the total. That is, if the frequencies in ascending order are $p_1, p_2, \ldots, p_n$, then p_ℓ is the frequency of the median symbol if

$$\sum_{i=1}^{\ell} p_i \geqslant \frac{1}{2}\sum_{i=1}^{n} p_i \text{ and } \sum_{i=\ell}^{n} p_i \geqslant \frac{1}{2}\sum_{i=1}^{n} p_i.$$

   At each step, your algorithm divides the current set of symbols into two parts, namely, those that are more (respectively, less) frequent than the median, and assigns the median to the "underdog" part, i.e., the part with the smaller total (breaking ties arbitrarily).

   Prove or disprove: The above algorithm returns an optimal prefix-free code.

8. ($\star$) Suppose you are given a connected graph $G = (V, E)$ with a cost c_e on each edge e. In class, we saw that when all edge costs are distinct, G has a unique minimum spanning tree. However, G may have many minimum spanning trees when the edge costs are not all distinct. Here we formulate the question: Can Kruskal's Algorithm be made to find *all* the minimum spanning trees of G?

   Recall that Kruskal's Algorithm sorted the edges in order of increasing cost, then greedily

processed the edges one by one, adding an edge e as long as it did not form a cycle. When some edges have the same cost, the phrase "in order of increasing cost" has to be specified a little more carefully: We'll say that an ordering of the edges is *valid* if the corresponding sequence of edge costs is nondecreasing. We'll say that a *valid execution* of Kruskal's Algorithm is one that begins with a valid ordering of the edges of G.

For any graph G, and any minimum spanning tree T of G, is there a valid execution of Kruskal's Algorithm on G that produces T as output? Give a proof or a counterexample.