COL351: Analysis and Design of Algorithms	Announced on: Aug 31, 2026 (Monday)

Tutorial Sheet 7

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 $\overline{S}$. 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 and 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 and 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 $G + e$. Provide a pseudocode and briefly justify (in one or two sentences) why the algorithm is correct and achieves the desired running time.

6. You wish to drive from point A to point B along a highway while minimizing the time you spend stopped for fuel. You are informed beforehand about the capacity C of your fuel tank in liters, your fuel consumption rate F in liters/kilometer, the refueling rate r in liters/minute at which you can fill your tank at a fuel station, and the locations $A = x_1, x_2, \ldots, x_{n-1}, x_n = B$ of the fuel stations along the highway. For example, if you stop to fill your tank from 2 liters to 8 liters, the stop would take $6/r$ minutes.

   Consider the following two strategies for refueling:

   a) Stop at every fuel station, filling the tank with just enough fuel to reach the next fuel station.

   b) Stop only if you don't have enough fuel to reach the next fuel station; if you do stop, fill the tank to its full capacity.

   For each strategy, either prove or disprove that it correctly solves the problem. If you are proving correctness, you must use an exchange argument.

7. Let $T = (V, E)$ and $T' = (V, E')$ be two different spanning trees of a graph G. Prove that there exists an edge $e \in E \setminus E'$ and an edge $e' \in E' \setminus E$ such that both $T \cup \{e'\} - \{e\}$ and $T' \cup \{e\} - \{e'\}$ are spanning trees of G.

8. ($\star$) Let $G = (V, E)$ be an undirected, unweighted, and connected graph with each edge colored either red or blue. Design an $\mathcal{O}(|E| \log |V|)$ algorithm that, given as input an integer k and the graph G, determines whether there exists a spanning tree of G that contains exactly k red edges.

9. ($\star$) Consider a connected graph $G = (V, E)$ where each edge e has an associated cost c_e. In class, we learned that if all edge costs are distinct, then G has a unique minimum spanning tree. However, when the edge costs are not all distinct, G may have multiple minimum spanning trees. This raises an interesting question: Can we modify Kruskal's algorithm to find *all* minimum spanning trees of G?

   Recall that Kruskal's algorithm sorts the edges in order of increasing cost and then greedily processes the edges one by one, adding an edge e as long as it does not form a cycle. When some edges have the same cost, the phrase "in order of increasing cost" needs to be specified more precisely. We define an ordering of the edges as *valid* if the corresponding sequence of edge costs is nondecreasing. A *valid execution* of Kruskal's algorithm is one that starts 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? Provide a proof or a counterexample.