CS 2443 : Examination

Department of Computer Science, IIT Hyderabad

May 1, 2022; Time – 5:00PM to 8:00PM

- Read the questions carefully and answer only to the questions.
- When asked to give an algorithm you need to prove correctness as well as analyse running time of your algorithm.
- Upload the answers and mark them towards the questions in gradescope. This time I will not accept answer sheets through mail. Please keep sufficient time to upload your answer sheets on gradescope and mark them.
- Maintain academic honesty.

---

1. Give the best possible asymptotic upper bounds for the following recurrence relations and prove your answer. The base case is $T(n) = 1$ when $n \leq 1$.

   (a) $T(n) = 2T(n/4) + 1$. **(2 marks)**

   (b) $T(n) = 3T(n/3) + n$. **(2 marks)**

2. We have seen a comparison based algorithm for finding median of n numbers with worst case running time $O(n)$. Prove that any comparison based algorithm for finding median of n numbers require at least $n - 1$ comparison. **(3.5 marks)**

3. A subsequence of a string $x_1 \ldots x_n$ is a string $y_1 \ldots y_r$ such that there exist $1 \leq i_1 < i_2 < \ldots < i_r \leq n$ with the property that for all $j \in \{1, \ldots r\}$, $y_j = x_{i_j}$.

   Write a recursive formula for the length of a longest common subsequence of three strings (1) $a_1 a_2 \ldots a_\ell$, (2) $b_1 b_2 \ldots b_m$, and (3) $c_1 c_2 \ldots c_n$. **(2.5 marks)**

4. Suppose you are given a undirected connected graph G with n vertices and m edges and edge costs are all distinct. A particular edge e of G is specified. Give an algorithm with running time $O(m + n)$ to decide whether e is contained in a minimum spanning tree of G. **(5 marks)**

5. Suppose you are given an undirected connected graph G, with edge costs that are all distinct. Prove that G has a unique minimum spanning tree. **(4 marks)**

6. Let us say that a graph $G = (V, E)$ is a *near-tree* if it is connected and has at most $n + 8$ edges, where $n = |V|$. Give an algorithm with running time $O(n)$ that takes a near-tree G with costs on its edges, and returns a minimum spanning tree of G. You may assume that all the edge costs are distinct. **(5 marks)**

7. Consider the Minimum Spanning Tree Problem on an undirected graph $G = (V, E)$, with a cost $c_e \geq 0$ 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. **(5 marks)**

8. In trying to understand the combinatorial structure of spanning trees, we can consider the space of all possible spanning trees of a given graph and study the properties of this space. This is a strategy that has been applied to many similar problems as well.

   Here is one way to do this. Let G be a connected graph, and T and T' two different spanning trees of G. We say that T and $'T$ are neighbors if T contains exactly one edge that is not in T', and T' contains exactly one edge that is not in T.

   Now, from any graph G, we can build a (large) graph $\mathcal{H}$ as follows. The nodes of $\mathcal{H}$ are the spanning trees of G, and there is an edge between two nodes of $\mathcal{H}$ if the corresponding spanning trees are neighbors.

   Is it true that, for any connected graph G, the resulting graph $\mathcal{H}$ is connected? Give a proof that $\mathcal{H}$ is always connected, or provide an example (with explanation) of a connected graph G for which $\mathcal{H}$ is not connected. **(5 marks)**

9. Suppose you are given a directed graph $G = (V, E)$, with a positive integer capacity c_e on each edge e, a designated source $s \in V$, and a designated sink $t \in V$. You are also given an integer maximum s-t flow in G, defined by a flow value f_e on each edge e. The flow f is *acyclic*: There is no cycle in G on which all edges carry positive flow. The flow f is also integer-valued.

   Now suppose we pick a specific edge $e^* \in E$ and reduce its capacity by one unit. Show how to find a maximum flow in the resulting graph in time $O(m + n)$, where m is the number of edges in G and n is the number of nodes. **(5 marks)**

10. Decide whether you think the following statement is true or false. If it is true, give a short explanation. If it is false, give a counterexample. **(3 marks)**

    > Let G be an arbitrary flow network, with a source s, a sink t, and a positive integer capacity c_e on every edge e; and let (A, B) be a mimimum s-t cut with respect to these capacities $\{c_e : e \in E\}$. Now suppose we add 1 to every capacity; then (A, B) is still a minimum s-t cut with respect to these new capacities $\{1 + c_e : e \in E\}$.

11. Suppose you and your friend live, together with $n-2$ other people, at a popular off-campus cooperative apartment. Over the next n nights, each of you is supposed to cook dinner for the co-op exactly once, so that someone cooks on each of the nights.

    Of course, everyone has scheduling conflicts with some of the nights (e.g., exams, concerts, etc.), so deciding who should cook on which night becomes a tricky task. For concreteness, let's label the people $\{p_1, \ldots, p_n\}$, the nights $\{d_1, \ldots, d_n\}$; and for person p_i, there's a set of nights $S_i \subsetneq \{d_1, \ldots, d_n\}$ when they are not able to cook.

    A feasible dinner schedule is an assignment of each person in the co-op to a different night, so that each person cooks on exactly one night, there is someone cooking on each night, and if p_i cooks on night d_j, then $d_j \notin S_i$.

    Describe a bipartite graph G so that G has a perfect matching if and only if there is a feasible dinner schedule for the co-op. **(3 marks)**

12. Let M be an $n \times n$ matrix with each entry equal to either 0 or 1. Let M_{ij} denote the entry in row i and column j.

    Swapping rows i and j of the matrix M denotes the following action: we swap the values M_{ik} and M_{jk} for $k = 1, 2, \ldots, n$. Swapping two columns is defined analogously.

    We say that M is *rearrangeable* if it is possible to swap some of the pairs of rows and some of the pairs of columns (in any sequence) so that, after all the swapping, all the diagonal entries of M are equal to 1.

    (a) Give an example of a matrix M that is not rearrangeable, but for which at least one entry in each row and each column is equal to 1.

    (b) Give a polynomial-time algorithm that determines whether a matrix M with 0-1 entries is rearrangeable.

    (5 + 5 marks)