CS204: Algorithms

End Semester, Spring 2016,

IIT Patna

<u>Please do not write anything on the question paper.</u>

Time: 3 Hrs | Full marks: 50

1. State true or false. No marks will be awarded without valid reasoning. Please try to answer these in the first two pages of your answer script. (1×10)

   (a) Let $G = (V, E)$ be a directed graph where every vertex has at most three outgoing edges. Then every vertex has at most three incoming edges.

   (b) Let $G = (V, E)$ be a directed graph. If $s \in V$ is a source and $t \in V$ is a sink, then there is a path from s to t.

   (c) If all of the edge capacities in a graph are an integer multiple of 7, then the value of the maximum flow will be a multiple of 7.

   (d) Is $n \lg n = \Theta(n^2)$?

   (e) Depth-first search on a connected undirected graph G will visit all of the vertices of G.

   (f) DFS on a directed graph with n vertices and at least n edges is guaranteed to find at least one back edge.

   (g) If we want to prove that a decision problem X is NP-complete, it is enough to reduce X to 3SAT (in other words, it is enough to prove $X \leq_P$ 3SAT).

   (h) If we can find a single problem in NP that has a polynomial-time algorithm, then there is a polynomial-time algorithm for 3SAT.

   (i) Every problem in P can be reduced to 3SAT.

   (j) If we have a dynamic programming algorithm with n^2 subproblems, is it possible that the running time could be asymptotically strictly more than $\Theta(n^2)$?

2. Answer briefly. (2.5×4)

   (a) Find asymptotic bound for $T(n)$ where $T(n) = 2T(n/2) + \frac{n}{\lg n}$

   (b) Perform a depth-first search on the graph (Fig - 1) starting at A. Label every edge in the graph with T if it is a tree edge, B if it is a back edge, F if it is a forward edge, and C if it is a cross edge. Whenever faced with a decision of which node to pick from a set of nodes, pick the node whose label occurs earliest in the alphabet.

   (c) Let $\mathcal{A}$ be an algorithm that solves the following problem. Given a set of integers $P = \{y_1, y_2, \ldots y_n\}$ $(y_i \geq 0)$, is it possible to divide the numbers into two disjoint sets (M, N say) such that sum of the numbers in both the sets are equal (that is $\sum_i m_i = \sum_i n_i$ where $m_i \in M$ and $n_i \in N$). Use algorithm $\mathcal{A}$ to solve the following problem. Given a set of integers $L = \{x_1, x_2, \ldots x_n\}$ $(x_i \geq 0)$ and an integer S, the algorithm finds a set $L' \subseteq L$ such that $\sum_i x'_i = S$ where $x'_i \in L'$.

   (d) Design an efficient algorithm to count the number of *shortest* paths from vertex s to vertex t in a directed graph G (possibly containing cycles; all edges are of length 1). (You can use any known algorithm as a black box)

3. You are given two strings, A and B. You need to find out the minimum cost of operations to transform A to B. Operations allowed are (a) delete one character from A (b) insert one character in A (c) replace a character of A by another character c. Each operation has a cost associated with it. (For example: Let A be FOOD and B be MONEY. Let all operations cost 1. The minimum cost for transforming A to B is 4. The operations are Replace F by M, Replace O by N, Replace D by E, Insert Y.)

   (a) Present an efficient *recursive* algorithm to find out the *minimum cost* to transform A to B.

   (b) Present a working example to demonstrate your algorithm.

   (c) Find complexity of your algorithm. $(3 + 2 + 2)$

4. Describe network flow problem and present an algorithm to find the maximum flow in a network. Present a working example using Fig. 2. Analyze time complexity of your algorithm. $(4 + 2 + 2)$

<u>Answer any 3 from the following.</u> (5×3)

5. Define Max-Clique problem. Prove that the decision version of the problem is NP-Complete.

6. Given a text $T[1, ..., n]$ (n characters) and a pattern $P[1, ..., m]$ (both of which are strings over the same alphabet), present a linear time algorithm to find all occurrences of P in T. Analyze time complexity of your algorithm.

7. If S is a set of vertices in an undirected graph $G = (V, E)$, define $f(S)$ to be the length of the shortest edge between a vertex in S and a vertex not in S, i.e., $f(S) = \min\{(v, w) : v \in S, w \notin S, (v, w) \in E\}$. Present an efficient algorithm for the following problem. Input: a connected, undirected graph G, with a non-negative length $l(e)$ on each edge e. Output: a non-empty set S that makes $f(S)$ as large as possible, subject to the requirement that $S \neq V$.

8. Given four vertices u, v, s and t in a directed weighted graph $G = (V, E)$ with non-negative edge weights, present an algorithm to find out if there exists a vertex $v_c \in V$ which is part of some shortest path from u to v and also a part of some shortest path from s to t. The algorithm should run in $O(E \log V)$. (You can use any known algorithm as a black box)

9. Prove that the average case time complexity of quick sort in $O(n \log n)$.

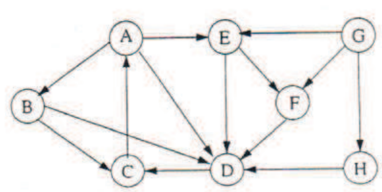

Fig - 1

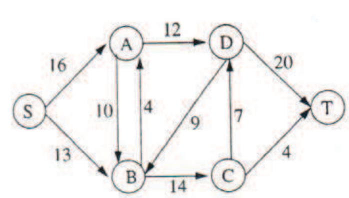

Fig - 2