CS204: Algorithms

End Semester, Autumn 2016,

IIT Patna

Please do not write anything on the question paper.

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) An algorithm whose running time satisfies the recurrence $P(n) = 1024 \times P(n/2) + O(n^{100})$ is asymptotically faster than an algorithm whose running time satisfies the recurrence $E(n) = 2 \times E(n - 1024) + O(1)$.

   (b) Given an undirected graph, it can be tested to determine whether or not it is a tree in $O(V + E)$ time. A tree is a connected graph without any cycles.

   (c) The Bellman-Ford algorithm applies to instances of the single-source shortest path problem which do not have a negative-weight directed cycle, but it does not detect the existence of a negative-weight directed cycle if there is one.

   (d) There exists a comparison-based algorithm to construct a BST from an unordered list of n elements in $O(n)$ time.

   (e) It is possible for a DFS on a directed graph with a positive number of edges to produce no tree edges.

   (f) In a top-down approach to dynamic programming, the larger subproblems are solved before the smaller ones.

   (g) Running a DFS on an undirected graph $G = (V, E)$ always produces the same number of cross edges, no matter what order the vertex list V is in and no matter what order the adjacency lists for each vertex are in.

   (h) If a problem in NP can be solved in polynomial time, then all problems in NP can be solved in polynomial time.

   (i) If an NP-complete problem can be solved in linear time, then all NP-complete problems can be solved in linear time.

   (j) For any flow network and any maximum flow on, there is always an edge such that increasing the capacity of increases the maximum flow of the network.

2. Answer briefly. (2.5×4)

   (a) Find exact solution for $T(n)$ where $2 \times T(n) = n \times T(n-1) + 3 \times (n!)$, and $T(0) = 5$.

   (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) Given a directed acyclic graph in which there is exactly one source node s and one sink node t. Give an efficient brief algorithm to find out the number of paths between s and t.

3. The Longest Increasing Subsequence (LIS) problem is to find the length of the longest subsequence of a given sequence such that all elements of the subsequence are sorted in increasing order. For example, length of LIS for $\{$ 10, 22, 9, 33, 21, 50, 41, 60, 80 $\}$ is 6 and LIS is $\{10, 22, 33, 50, 60, 80\}$.

   (a) Present an efficient *recursive* algorithm to find out the *length of longest subsequence* and the *subsequence.*

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

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

4. Describe Kruskal's algorithms to find a minimum spanning tree of a given undirected graph. Analyze the time complexity of the algorithm. Present a working example using Fig - 2. $(4 + 2 + 2)$

Answer any 3 from the following. (5×3)

5. Define 3-SAT problem. Prove that 3-SAT 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. A set of cities V is connected by a network of roads $G(V, E)$. The length of road $e \in E$ is l_e (≥ 0). There is a proposal to add one new road to this network, and there is a list E' of pairs of cities between which the new road can be built. Each such potential road e' has an associated length. As a lobbyist for city s, you wish to determine the road $e' \in E'$ that would result in the maximum decrease in the driving distance between s and a particular city t. Give an efficient brief algorithm for solving this problem, and analyze its running time as a function of $|V|$, $|E|$ and $|E'|$.

8. Present an algorithm to sort n integers in the range of 1 to $(n^2 - 1)$ in $O(n)$ time.

9. Prove that the average case time complexity for construction of binary search tree using n keys is $O(n \log n)$.

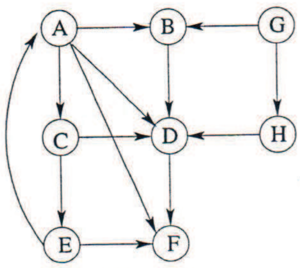

Fig - 1

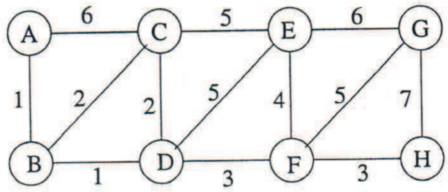

Fig - 2