CS514: Design and Analysis of Algorithms

End Semester, Spring 2017,

IIT Patna

Please do not write anything on the question paper.

Time: 3 Hrs

Full marks: 50

1. Answer briefly. No marks will be awarded without valid reasoning. (1×10)

   (a) Comment on worst case running time and expected running time of Rabin-Karp string matching algorithm. Assume the the length of the text we are looking into is n and length of the pattern that we are looking for is m.

   (b) For a complete graph, cost of a Hamiltonian cycle is always lower than the cost of any spanning tree that we can find. Comment with justification.

   (c) It is always better to use merge sort instead of quick sort as merge sort guarantees $O(n \log n)$ running time. Comment with justification.

   (d) Dijkstra's shortest path algorithm for a single source always include edges which should be included in minimum cost spanning tree. Comment with justification and example.

   (e) In some special case maximum amount of flow can be more than minimum cut. Comment with justification.

   (f) The depths of any two leaves in a max heap differ by at most 1.

   (g) Every directed acyclic graph has exactly one topological ordering.

   (h) Consider a weighted directed graph $G = (V, E, w)$ and let X be a shortest $s-t$ path for $s, t \in V$. If we double the weight of every edge in the graph, setting $w(e) = 2w(e)$ for each $e \in E$, then X will still be a shortest $s - t$ path in (V, E, w).

   (i) If a depth-first search on a directed graph $G = (V, E)$ produces exactly one back edge, then it is possible to choose an edge $e \in E$ such that the graph $G = (V, E - \{e\})$ is acyclic.

   (j) Consider two positively weighted graphs $G = (V, E, w)$ and $G' = (V, E, w')$ with the same vertices V and edges E such that, for any edge $e \in E$, we have $w'(e) = w(e)^2$. For any two vertices $u, v \in V$, any shortest path between u and v in G' is also a shortest path in G.

2. Answer briefly. (2.5×4)

   (a) Find asymptotic complexity of an algorithm with runtime as $T(n) = \sqrt{n}T(\sqrt{n}) + 2017n$

   (b) Perform a depth-first search on the graph (Fig - 2) 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 $P = \{y_1, y_2, \dots y_n\}$ $(y_i \geq 0)$, 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 $L = \{x_1, x_2, \dots x_n\}$ $(x_i \geq 0)$ 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. State maximum flow problem. Present an efficient algorithm to solve this problem. Analyze time complexity of your algorithm. Run your algorithm on the graph shown in Fig 1 (Please show (a) residual graph at the end of first iteration, (b) final flow). (2+4+2+2)

Answer any 4 from the following. (5 × 4)

4. Give a 2-approximation algorithm for traveling salesman problem.

5. An independent set of a graph $G = (V, E)$ is a subset V' of V such that each edge in E is incident on at most one vertex in V'. The independent set problem is to find a maximum size independent set in G. Formulate a related decision problem for the independent set problem and prove it is NP-Complete.

6. String matching: 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 the time complexity of your algorithm.

7. Give an efficient algorithm to find out transitive closure of a directed graph. Analyze complexity of your algorithm.

8. Given a list of N coins, their values $(V_1, V_2, \ldots, V_N)$, and the total sum S. Provide an efficient dynamic programming algorithm to find the minimum number of coins the sum of which is S (you can use as many coins of one type as you want), or report that it's not possible to select coins in such a way that they sum up to S. Analyze time complexity of your algorithm.

9. In the linear time algorithm of finding the k-th smallest element out of n elements, the input elements are divided into groups of 5. Will the algorithm work in linear time if they are divided into groups of (a) 7, (b) 3? Justify.

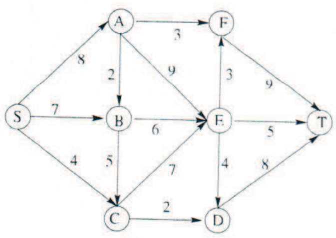

Fig - 1

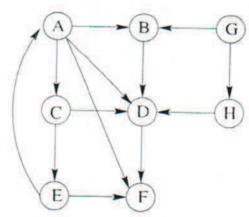

Fig - 2