Bengal Engineering and Science University, Shibpur
B.E. 5th Semester Examination, 2011
Design and Analysis of Algorithms (CS-504)

Time: 3 hours Full Marks: 70

Attempt any FIVE questions.

1.a) Prove that, for any two functions $f(n)$ and $g(n)$, $f(n) = \Theta(g(n))$ if and only if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$.

b) Write an $O(n)$ running time algorithm to build a heap of an array of size n and illustrate its operations using the array A = {5, 3, 17, 10, 84, 19, 6, 22, 9}. Write not only the algorithm but also compute its running time. (6+8)

2.a) Let $a \geq 1$ and $b > 1$ be constants and $f(n)$ be a nonnegative function defined on exact powers of b. A function $g(n)$ defined over exact powers of b by

$$g(n) = \sum_{j=0}^{\log_b n - 1} a^j f(n/b^j)$$

Prove that, if $af(n/b) \leq c\, f(n)$ for some constant $c < 1$ and all $n \geq b$ then $g(n) = \Theta(f(n))$.

b) Use substitution method to solve the following recurrence equation:

$$T(1) \geq 1,$$

$$T(n) \geq 1 + \sum_{k=1}^{n-1} (T(k) + T(n-k) + 1) \text{ for } n > 1.$$

c) Draw the recursion tree for $T(n) = T(n/3) + T(2n/3) + n$, and provide a good asymptotic upper bound on its solution. (5+5+4)

3.a) Prove that the theoretical minimum lower bound of time complexity for any sorting algorithm where the sorting is performed by pair wise comparison is $O(n \log_2 n)$.

b) Give sufficient justification by which Radix Sort can be considered as a Bucket Sort. (7+7)

4.a) Compare and contrast the techniques commonly used for generating probe sequences in Open Addressing Scheme.

b) Assuming uniform hashing, prove that for an open-address hash table with load factor $\alpha < 1$, the expected number of probes in a successful search is at most $\frac{1}{\alpha} ln \frac{1}{1-\alpha} + \frac{1}{\alpha}$. (7+7)

5.a) What are the key ingredients of an optimization problem that make it suitable for solution by the Dynamic programming strategy? In a Matrix-Chain Multiplication problem, stated below, show that all the above ingredients exist.

Matrix-Chain Multiplication problem: Given a chain $\langle A_1, A_2, \ldots, A_n \rangle$ of n matrices, where for $i = 1, 2, \ldots, n$, matrix A_i has

dimension $P_{i-1} \times P_i$, fully parenthesize the product $A_1A_2...A_n$ in a way that minimizes the number of scalar multiplications.

b) Write an algorithm using Dynamic Programming Strategy to solve the above problem and compute its running time. (7+7)

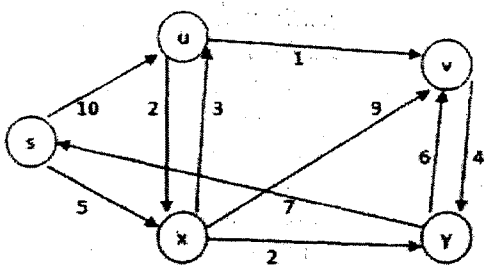
6.a) An "Activity selection problem" is the problem of scheduling a resource among several competing activities. Write a greedy algorithm for selecting maximum-size set of mutually compatible activities, stating the assumptions (if any) made for it.

Suppose you have a set S = {1, 2, 3, …, 11} of 11 activities that wish to use a resource. Say, S_i and f_i are the start time and finish time of i-th activity as shown in the following table. Compute a maximum-size set of mutually compatible activities by using the algorithm.

I	1	2	3	4	5	6	7	8	9	10	11
S_i	8	5	0	8	3	1	12	5	3	6	2
f_i	11	9	6	12	8	4	14	7	5	10	13

b) Write the Kruskal's algorithm to construct the Minimum Spanning Tree of a connected, undirected graph. Use disjoint-set data structure. Discuss its running time for both dense and sparse graph. (7+7)

7.a) Illustrate the Dijkstra's algorithm on the following graph G = (V, E) and find the shortest path from source vertex *s* to all other vertices.



b) For the correctness of the Dijkstra's algorithm, prove that if the algorithm runs on a weighted, directed graph G = (V, E) with nonnegative weight function w and source *s*, then at termination, d[u] = $\delta(s, u)$ for all vertices $u \in V$. Here, d[u] is the weight of the estimated shortest path or an upper bound on the weight of a shortest path from source *s* to *u* and $\delta(s, u)$ is the weight of the shortest path from *s* to *u*. (6+8)

8.a) What do you mean by Polynomial time reducibility of a problem to another problem? When a problem is called NP-hard problem?

b) If a problem A is polynomial time reducible to a problem B and A is a problem of NP-complete class then prove that B is a problem of NP-hard class.

c) Define the vertex-cover problem and prove that it is a problem of NP-complete class. (3+3+8)