C4-R3: ALGORITHM ANALYSIS AND DESIGN

NOTE:

1.	**Answer question 1 and any FOUR questions from 2 to 7.**
2.	**Parts of the same question should be answered together and in the same sequence.**

Time: 3 Hours **Total Marks: 100**

1.

a) Let *X* be a singly linked list and *P* be an intermediate node. Write an O(1) algorithm to delete the node *P* from the list.

b) Consider the classical $O(n^3)$ matrix multiplication algorithm to multiply two $n \times n$ matrices. Observe that there are two operations *addition* (+) and *multiplication* (*). In order to determine all pairs of shortest paths for a connected graph with *n* vertices, this matrix multiplication algorithm can be used but one needs to replace these two operations by another two operations. Mention the name of operation to be used as a replacement of *addition* and *multiplication*.

c) Define the meaning of *Parallel Cost Optimal Algorithm*.

d) Draw all the binary trees having at most 3 nodes.

e) Determine the minimum number of comparisons needed to compute the minimum and the maximum of *n* numbers.

f) Show that for any real constants a and b, where b>0 $(n+a)^b=\theta(n^b)$.

g) Mention a strategy that can be adopted to show that a problem *X* is NP-hard.

(7x4)

2.

a) Discuss diagrammatically the various cases that can occur at the time of deleting a node in Red-Black tree. Illustrate with the help of a suitable example.

b) Write an algorithm to determine for every pair of vertices the existence of a path of length at most *k* from vertex *i* to vertex *j*.

(9+9)

3.

a) Write an algorithm to compute convex hull of the *n* input points which makes use of stack. What is its time complexity?

b) Write the Boyer-Moore String Matching algorithm. What is its time complexity? Mention the situations when one should use the naive algorithm, Boyer-Moore algorithm and Knuth-Morris-Pratt algorithm.

(8+10)

4.

a) Write a cost optimal parallel algorithm to find the sum of *n* numbers.

b) Given a flow network $G = (V,E)$, let f_1 and f_2 be functions from $V\times V$ to $\Re$. The flow sum $f_1 + f_2$ is the function from $V\times V$ to $\Re$ defined by

$$(f_1+f_2)(u,v) = f_1(u,v) + f_2(u,v)$$

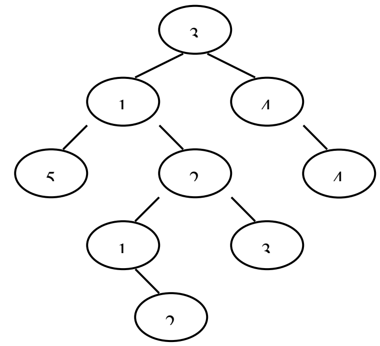
For all $u,v \in V$. If f_1 and f_2 are flows in *G*, show that the flow sum $(f_1+ f_2)$ satisfies the skew symmetry and flow conservation properties but may violate the capacity constraint property.

(9+9)

5.

a) Write an efficient algorithm to determine the second largest element of *n* numbers. Determine the number of comparisons needed to find it.

b) Consider the following binary tree. Put the balance factor against each node. Is it an AVL tree? If not, take the necessary step to balance it.



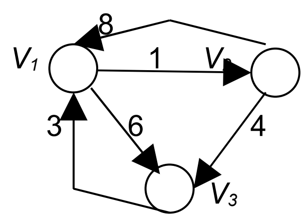
Perform the following operations and at the end of each operation, balance the tree in case it has lost the property of the AVL tree. Insert (32), Insert (50), Delete (20).

(9+9)

6.

a) Let there be a set of four identifiers $\{a_1, a_2, a_3, a_4\}$ = {*do, end, stop, while*}. Let us define five classes of identifiers $E_0, E_1, E_2, E_3,$ and E_4 where E_i contains all identifiers X satisfying $X < a_i$, i = 0, 1, 2, 3 and E_4 consists of all identifiers X satisfying $X > a_4$. Let p_i be the probability for a successful search of the identifier a_i, i = 1, 2, 3, 4 and q_i be the probability that the identifier being searched is in class E_i, i =0, 1, 2, 3, 4. Obtain the Optimal Binary Search Tree for the identifier set $\{a_1, a_2, a_3, a_4\}$ having $p_1 = 0.1$, $p_2 = 0.2$, $p_3 = 0.2$, $p_4 = 0.1$, $q_0 = 0.1$, $q_1 = 0.1$, $q_2 = 0.1$, $q_3 = 0.05$ and $q_4 = 0.05$.

b) For the following graph G, determine the matrix C where $(i,j)^{th}$ element C_{ij} gives the cost of the shortest path from V_i to V_j, for all i and j.



(10+8)

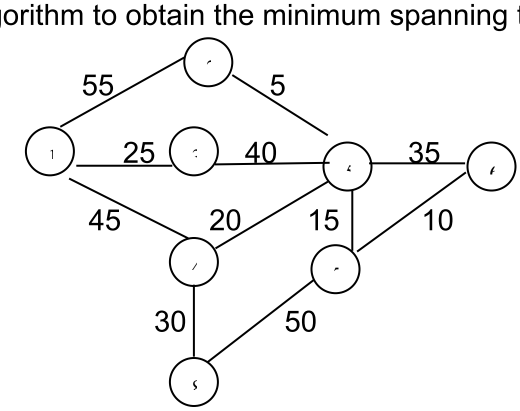
7.

a) Let A and B be two n-bit numbers where n is a power of 2. A divide-and-conquer based algorithm divides the numbers into two equal parts to compute the product as $AB = A_1 B_1 2^n + (A_1 B_2 + A_2 B_1) 2^{n/2} + A_2 B_2$ where $A = (A_1 2^{n/2} + A_2)$ and $B = (B_1 2^{n/2} + B_2)$ and A_1B_1, A_1B_2, A_2B_1 and A_2B_2 are computed recursively. Then,

i) Determine the time complexity.

ii) If $(A_1 B_2 + A_2 B_1)$ is computed as $(A_1 + A_2)(B_1 + B_2) - A_1 B_1 - A_2 B_2$ then will there be any change in the time complexity? Justify.

b) Trace the Kruskal's algorithm to obtain the minimum spanning tree from the following graph.



(10+8)