29 NOV 2018

B. Tech IT 3rd Semester Nov/Dec 2018
ITPC-21 Design & Analysis of Algorithms

Time: 3 hours | Total Marks: 50

Note: 1. Questions 1, 4, 5, 6 are compulsory. Attempt one out of Question 2 & 3
2. If you develop some programs using some special logic, provide necessary comment/explanation for that.
3. While writing any module/function, if it calls any other function, also write the code of that function. If your code uses some structures and global variables, define those also. Use of STL, APIs is not allowed.
4. In all programming oriented questions, efficiency of the algo & modularity will also be a criteria of evaluation along with the correctness.
5. Think logically. Concepts/algos/notations described in your text book of Algorithms are to be used, wherever unspecified.

1. (a) Can Master Theorem be applied to the Recurrence $T(n) = 4T(n/2)+n^2\log n$? Why or why not? Compute an asymptotic upper bound for this recurrence. **4**

   (b) With ref to heap sort, consider data (6, 4, 18, 11, 15, 20, 7, 24, 10, 86) to be used as input for the function BUILD-MAX-HEAP. Show the initial heap. BUILD-MAX-HEAP works iteratively by heapifying one element at a time. Show the **heap after each step of BUILD-MAX-HEAP in which an exchange happens**. Don't write the code. **3**

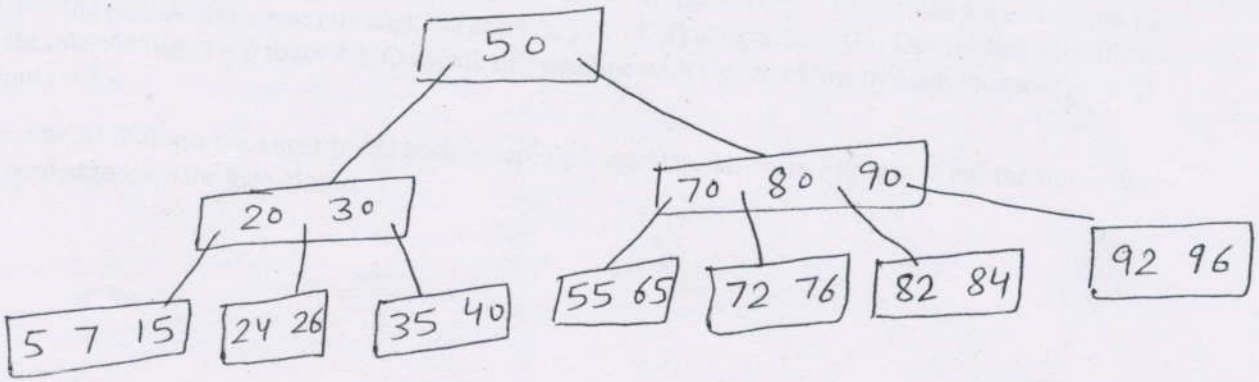
   (c) Given root of a binary search tree, write a function in C which counts total number of non-leaf nodes. Don't use any global/static variables for this. **3**

2. (a) For a **B-tree**, you are given node **x**, node **y** and index **i**. Here y is ith child of x. and is the node to be split. Node y contains 2t-1 keys. During the insertion logic, it has been identified that node y needs a split. Now **write a function SPLITCHILD in C,** which splits node y. **write suitable comments** inside the code to support your logic. Also mention **when and why DISK READ/WRITE** is needed during this function? **7**

   (b) Assume that alphabets (A,B,C,D,E) have frequencies 10, 3, 8, 25, 9 respectively. Apply Huffman coding algo on this and show all steps of Huffman tree generation. Then show the Huffman code generated for each alphabet. No need to write the code/algo. **3**

OR

3. (a) Consider the following B-Tree of degree t=3. Show B Tree after deletion of following values one by one: 72, 80, 24, 50, 5 **5**



(b) With reference to Huffman code's greedy algorithm, prove the following Lemma: Let C be an alphabet in which each character c € C has freq c.freq. Let x and y be two characters in C having the lowest frequencies. Then there exists an optimal prefix code for C in which the codewords for x and y have also the same length and differ only in the last bit. 5

4. (a) The activity selection problem can be solved using Dynamic Programming also. Write steps of developing its recursive solution first and then write its recursive code/algo. Now write its memorized code/algo. 7

(b) Write a **C function** for finding backtracking solution of m-coloring problem. Assume graph is stored as adjacency matrix. Don't use any global variables. 3

5. (a) Consider the graph of Fig 5 (a) and step by step (show values after each main step), compute its Strongly Connected Components. No need to write the code. Traverse the vertices in ascending order of their names, whenever there are multiple options. 5

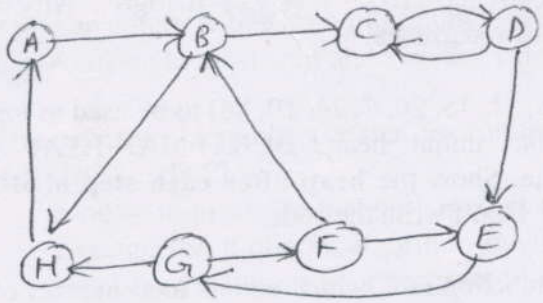


Fig 5(a)

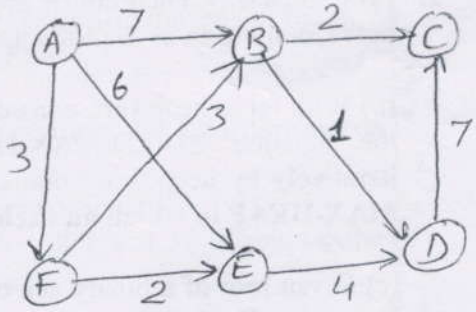


Fig 5(b)

(b) Consider the graph of Fig 5 (b) and step by step (show all computed values after each main step), compute shortest paths from source vertex A to all vertices using Dijkastra's Algorithm. No need to write the code. 5

6. (a) What are problems of primary clustering and secondary clustering? In which kind of synonym resolutions, these problems occur? 4

(b) Write the algo/code of counting sort and mention its time complexity. 3

(c) Write a **C recursive function** to generate greedy solution of the activity selection problem. Assume that list of activities is already sorted as per your needs. The list of selected jobs should be stored. 3

END