Total No. of Pages:3

IIIrd SEMESTER

END SEMESTER EXAMINATION

Roll No.....

B.Tech.(COE/SE)

(NOV. – 2018)

Paper Code: CO-201

Time: 3:00 Hours

Title: Data Structures

Max. Marks: 40

Note: 1. Attempt all questions. Assume any suitable value(s) for missing data.
2. If asked to write algorithms, write as C functions or pseudo codes.
3. If any assumption is made for any question, write it.

1. (a) Consider an array A of integers. Suppose that A.length returns the length of such an array.

   (i) Write down an iterative algorithm in pseudocode that returns true if the array is sorted and false otherwise.

   (ii) Write a recursive algorithm following the divide-and-conquer approach (where problem size is reduced in each recursive call) that tests whether A is sorted. Start from the fragment below.

   (4+4=8)

2. (a) A connected component of a graph is a set of vertices where each node can reach every other node in the component along the given edges, and which is connected to no additional vertices. Give a modified version of Kruskal's algorithm to compute the number of connected components in a graph.

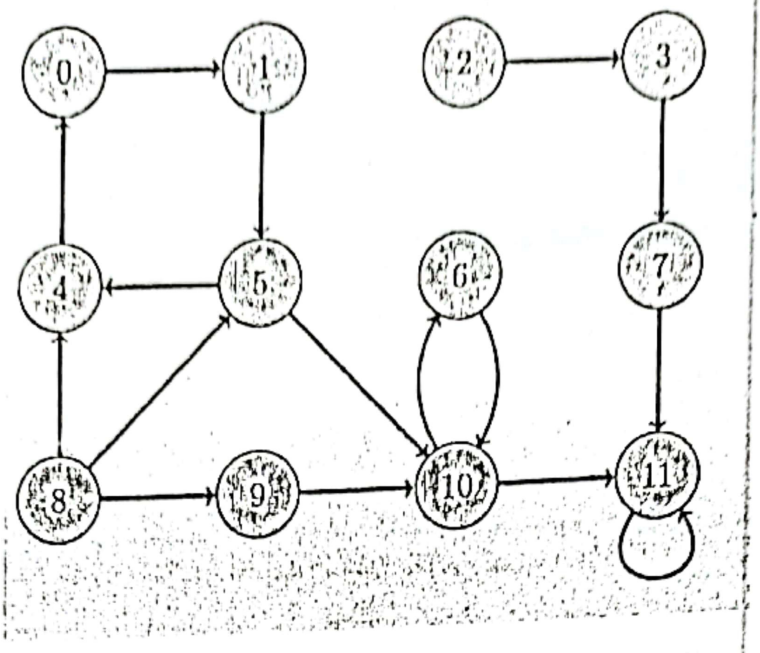
   (b) Suppose that we have numbers between 1 and 100 in a binary search tree and want to search for the number 45. Which (possibly multiple) of the following sequences could be the sequence of nodes examined?

   - 5, 2, 1, 10, 39, 34, 77, 63.
   - 1, 2, 3, 4, 5, 6, 7, 8.
   - 9, 8, 63, 0, 4, 3, 2, 1.
   - 8, 7, 6, 5, 4, 3, 2, 1.
   - 50, 25, 26, 27, 40, 44, 42.
   - 50, 25, 26, 27, 40, 44.

   (4+4=8)

3. (a) Suppose we have a MaxHeap H and two values v_1 and v_2, such that all values are distinct. Let H_{12} be the heap you get if you insert v_1 and then v_2 into H, and H_{21} be the heap you get if you insert v_2 and then v_1 into H. Give an example of H, v_1 and v_2 such that $H_{12} \neq H_{21}$. No justification needed, just draw the heaps H, H_{12} and H_{21}.

   (b) Give one possible DFS traversal starting from node 0 of the graph below, printing the nodes both as they are discovered and finished. Also classify all edges into 4 classes (tree, Back, forward, cross). Note that the graph is directed.



(4+4=8)

4. (a) Since storing a triangular matrix as a two dimensional array wastes space, we would like to find a way to store only the nonzero terms in the triangular matrix. Find an addressing formula for the elements $L_{i,j}$ to be stored in a one dimensional array a. In other words, where in the array a would you store the element $L_{i,j}$? (You only need to explain how to map each nonzero element to the one dimensional array. No need to write down the actual function.)

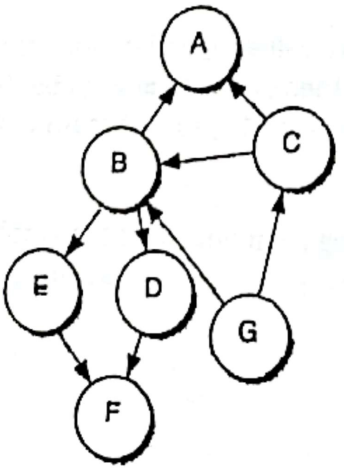
(b) Given an array of N elements of three different types: Cold, Warm, and Hot, design and describe clearly an O(N) in-place algorithm to put all the cold elements, on the left, followed by all the warm elements, followed by all the hot elements on the right. Your algorithm can use only a small **constant amount of extra space.** Show how your algorithm would operate on following array:

0	1	2	3	4	5	6	7	8	9	10
C	W	H	C	W	W	C	H	C	W	C

(4+4=8)

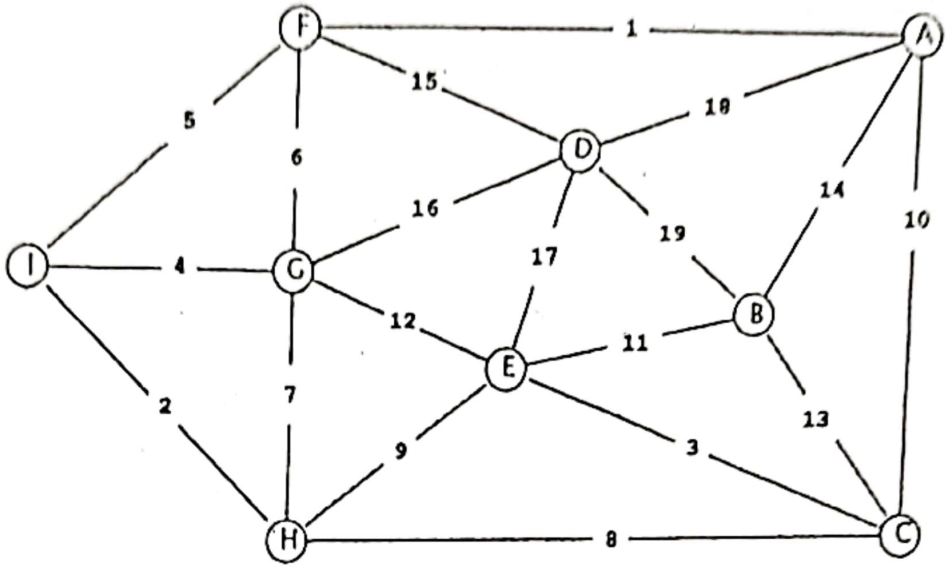
5. (a) Draw the B-trees that result when adding the following values in succession, starting with an empty tree. Assume each node can only hold 2 keys. To save drawing time, you can choose to only draw a new tree when a split occurs, but make it clear which value caused the split. Values: 35, 87, 64, 78, 81, 85, 22, 31.

(b) Find a topological ordering of the following graph?

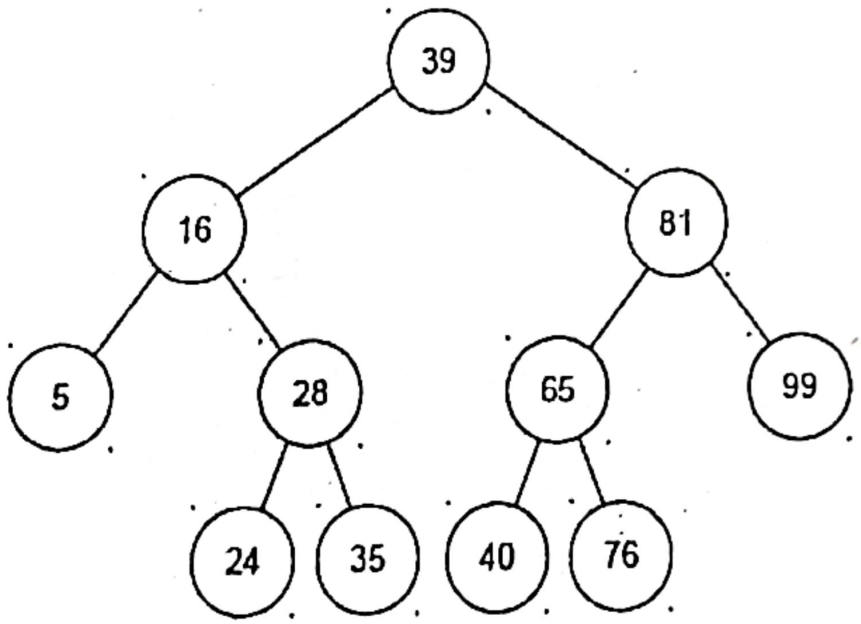


(4+4=8)

6. (a) Compute MST for the given graph using Prim's algorithm showing graph after each iteration of the algorithm.



(b) Consider following AVL tree. Draw AVL tree after inserting a new key 25.



(5+3=8)

7. (a) Given two arrays a[] and b[], each containing n distinct numbers, design two algorithms (with different performance) to determine whether the two arrays contains precisely the same set of numbers (but possibly in a different order).

(b) Write insertion sort algorithm for sorting a linked list into ascending order.

(4+4=8)

8. (a) Write algorithm count_LEAF() to count number of leaf nodes in a given binary tree.

(b) Write an algorithm isBST() to check if given binary tree is a Binary Search Tree.

(4+4=8)