Reg.No:							

SNS College of Technology, Coimbatore-35.
(Autonomous)
B.E/B.Tech- Internal Assessment -II
Academic Year 2023-2024(ODD)
Third Semester
19ITT201- Data Structures
(Common to CSE,IT and AIML)

A

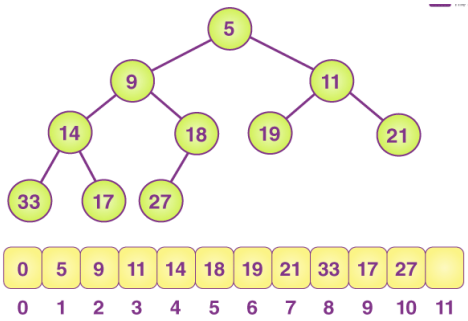
Time: $1^{1/2}$ Hours **Maximum Marks: 50**

Answer All Questions

PART-A (5 x 2 = 10 Marks)

1. Illustrate how Binary Heap Represented using array with an example. CO2 Ana

A Binary Heap is like a complete Binary Tree. It is commonly represented as an array.



2. Define Huffman tree and its application. CO2 Und

The Huffman tree is treated as the binary tree associated with minimum external path weight that means, the one associated with the minimum sum of weighted path lengths for the given set of leaves. So the goal is to construct a tree with the minimum external path weight.

3. Distinguish directed graph and undirected graph CO3 Ana

A **directed graph** is a set of **vertices** (nodes) connected by **edges**, with each node having a direction **associated** with it. Edges are usually represented by arrows pointing in the direction the graph can be traversed.

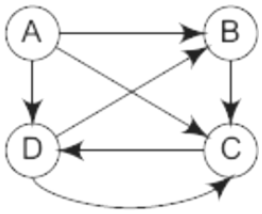
Undirected Graphs

In an **undirected graph** the edges are bidirectional, with no direction associated with them. Hence, the graph can be traversed in either direction. The absence of an arrow tells us that the graph is undirected.

4. What is the difference between an Euler path and a circuit? CO3 Ana

A path in a graph is a succession of adjacent edges, with no repeated edges, that joins two vertices. Definition. A circuit is a path which joins a node to itself

5. Find out the in degree and out degree of each node in the given graph. CO3 App



Node	In degree	Out degree
A	0	3
B	2	1
C	2	1
D	2	2

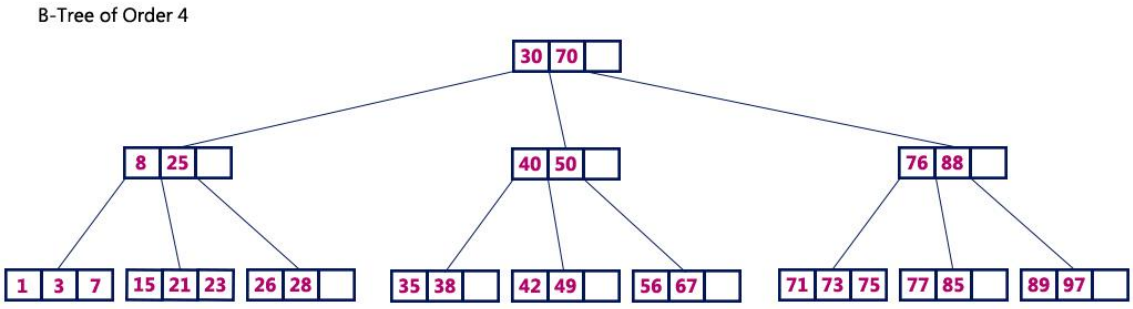
PART–B (13+13+14 = 40 marks)

6. (a) Explain in detail about the B tree construction and its operations with a neat illustration. 13 CO2 Und

B-Tree is a self-balancing search tree.

B-Tree of Order m has the following properties...

- Property #1 - All leaf nodes must be at same level.
- Property #2 - All nodes except root must have at least [m/2]-1 keys and maximum of m-1 keys.
- Property #3 - All non-leaf nodes except root (i.e. all internal nodes) must have at least m/2 children.
- Property #4 - If the root node is a non leaf node, then it must have atleast 2 children.
- Property #5 - A non leaf node with n-1 keys must have n number of children.
- Property #6 - All the key values in a node must be in Ascending Order.

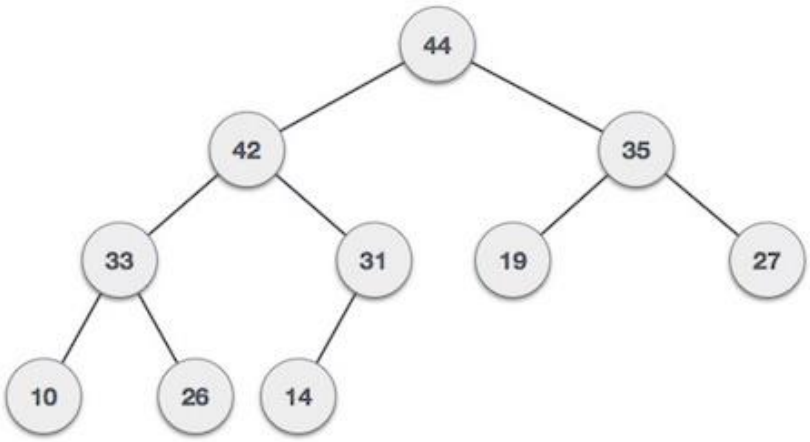


(or)

(b) Show the result of inserting 33,35,42,10,14,19,27,44,26,31 one at a time, into an initially empty binary heap. Also show the result of performing two delete Min operations in the final binary heap obtained. 13 CO2 App

Insertion algorithm

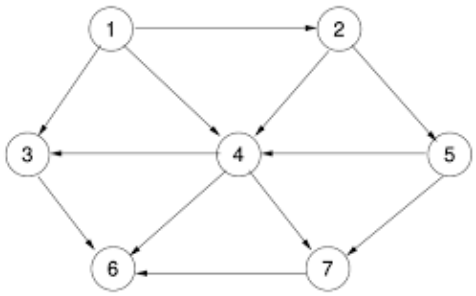
Step 1 − Create a new node at the end of heap.
Step 2 − Assign new value to the node.
Step 3 − Compare the value of this child node with its parent.
Step 4 − If value of parent is less than child, then swap them.
Step 5 − Repeat step 3 & 4 until Heap property holds.



Deletion algorithm

Step 1 − Remove root node.
Step 2 − Move the last element of last level to root.
Step 3 − Compare the value of this child node with its parent.
Step 4 − If value of parent is less than child, then swap them.
Step 5 − Repeat step 3 & 4 until Heap property holds.

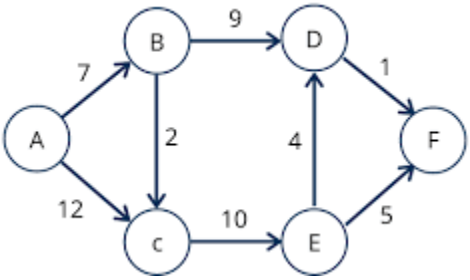
7. (a) Find the topological sort for the given graph using queue with algorithm. 13 CO3 App

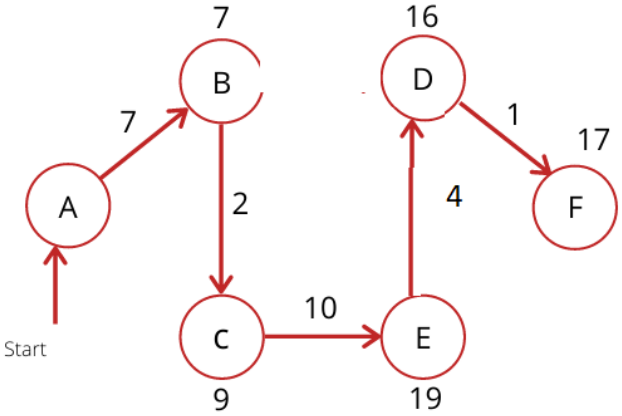


Order of Visit : 1,2,5,4,3,7,6

(or)

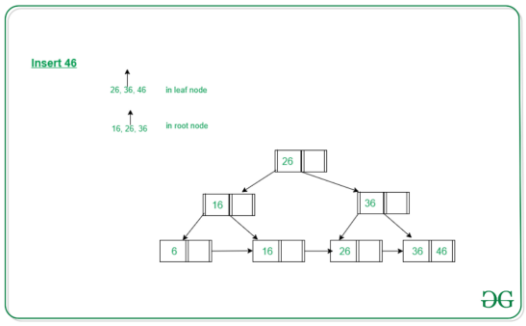
(b) Find the shortest path from "**A**" to all other vertices for the given graph using Dijkstra's algorithm. 13 CO2 App





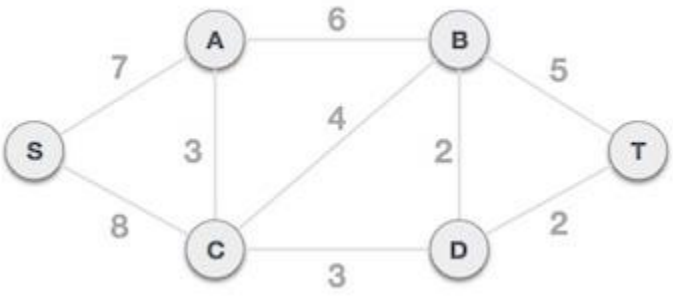
8. (a) Explain about the B+ trees and Insert the following key values 6, 16, 26, 36, 46 on a B+ tree with order = 3. 14 CO2 App

- Each node except root can have a maximum of M children and at least ceil(M/2) children.
- Each node can contain a maximum of M – 1 keys and a minimum of ceil(M/2) – 1 keys.
- The root has at least two children and atleast one search key.
- While insertion overflow of the node occurs when it contains more than M – 1 search key values.



(or)

(b) Apply Kruskal's algorithm and find the minimum spanning tree for the given graph. 14 CO3 App

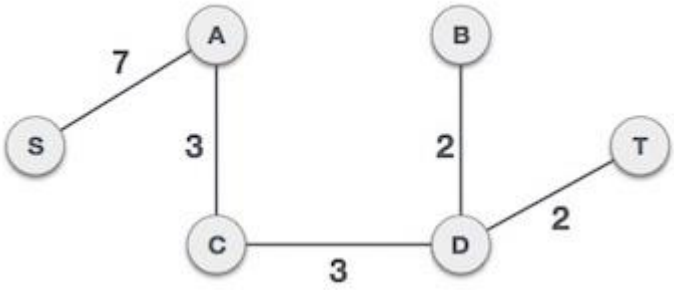


Arrange all edges in their increasing order of weight

The next step is to create a set of edges and weight, and arrange them in an ascending order of weightage (cost).

B, D	D, T	A, C	C, D	C, B	B, T	A, B	S, A	S, C
2	2	3	3	4	5	6	7	8

Add the edge which has the least weightage and confirm the edges if it doesn't form the cycle



(Note: Und-Understand Rem-Remember App-Apply Ana-Analyze)

Prepared by **Verified By** **HoD**