Unit1: Introduction toTree

Mapped course outcome: CO1

Low level of difficulty questions :

1. Describe the following tree terminologies with an example:
   - Root, Parent, Child, Sibling, Leaf, Internal Node, Height, Depth and Degree.

2..Distinguish the tree and graph with at least three key differences and relevant examples.

3. Elucidate the tree representation using an array with an example.

4. Describe the tree representation using a linked list with an example.

5. Compare the binary tree and binary search tree(BST) with an examples.

6. Explain the properties of a Binary Tree with an example.

7. Write a function to count the number of leaf nodes in a binary tree recursively.

8. Write a function to find the height(maximum depth)of a binary tree.

9. Write a function to count the total number of nodes in a binary tree.

10. Explain Recursive and Non-Recursive approaches for finding the height of a binary tree.

Medium level of difficulty questions :

11. Illustrate an algorithm for recursive inorder,preorder and postorder traversals of a binary tree.

12. Describe an algorithm for iterative inorder,preorder,and postorder traversals using stacks.

13. Compare the recursive and iterative tree traversals in terms of space and time complexity.

14. Construct the binary tree and draw its structure for the below given traversals

Inorder:[D,B,E,A,F,C,G] and Preorder:[A,B,D,E,C,F,G]

15. Write an algorithm to construct a binary tree using inorder and postorder traversals.

16. Write an algorithm to construct a binary tree using inorder and preorder traversals.

17. Construct a BinaryTree from given traversals and explain the steps:

- Inorder: [4, 2, 5, 1, 6, 3, 7]
- Postorder: [4, 5, 2, 6, 7, 3, 1]

18. Explain the process of constructing a binary tree from traversal data and discuss its time complexity.

19. Write an algorithm for inserting a node in a Binary Search Tree(BST).

20. Write an algorithm for deletion in a BST, covering different cases(leaf node,one child, two children).

High level of difficulty questions :

21. Analyse the algorithm for searching a node in a BST along with its time complexity.

22. Construct a Binary Search Tree(BST) for the sequence:[50,30,70,20,40,60,80] and perform inorder ,pre order traversals.

23. Elucidate the BST as an Abstract Data Type (ADT) with its key operations.

24. Analyse the advantages and limitations of Binary Search Trees(BST).

25.Represent a company's organizational structure using an array-based binary tree and retrieve the CEO's direct subordinates.

26. Implement a binary tree using a linked list and write an algorithm to find the height of the tree.

27.Write an algorithm to count the number of leaf nodes recursively for a binary tree with n nodes.

28. A sports tournament is represented as a binary tree where each match is a node.Write a function to count the total number of matches.

29. Compare the space complexity of recursive and iterative inorder traversal of a binary tree with 10,000 nodes.

30. Construct a binary tree for the given inorder traversal [D,B,E,A,F,C,G] and preorder

traversal [A,B,D,E,C,F, G]

31. Insert the elements {55, 30, 70, 20, 40, 60, 80} into a BST and show its structure.

32. Given a BST with the elements{50,30,70,20,40,60,80},delete the node 30 and show the updated tree.

33. Implement an algorithm to find whether the value 75 exists in a BST with n nodes and analyze its time complexity.

34. Discuss how a BST can be used in an auto complete search feature where words are stored in alphabetical order.

Previous GATE exam questions :

1. The post-order traversal of a binary tree is ACEDBHIGF. The pre-order traversal is

a]ABCDEFHI
b]FBADCEIH
c]FABCDEGI
d]ABDCEFGI

2. The height of a tree is the length of the longest root-to-leaf path in it.The maximum and minimum number of nodes in a binary tree of height 5 are

A] 63 and 6, respectively

B] 64 and 5, respectively

C] 32 and 6, respectively

D] 31 and 5, respectively

3.The preorder traversal of a binary search tree is 15,10,12,11,20,18,16,19. Which one of the following is the postorder traversal of the tree?

A]10,11,12,15,16,18,19,20

B]11,12,10,16,19,18,20,15

C]20,19,18,16,15,12,11,10

D]19,16,18,20,11,12,10,15

4. While inserting the elements 71,65,84,69,67,83 in an empty binary search tree(BST)in

the sequence shown, the element in the lowest level is

A] 65 B] 67 C] 69 D] 83

5. The preorder traversal sequence of a binary search tree is 30,20,10 ,15, 25, 23, 39, 35,42. Which one of the following is the postorder traversal sequence of the same tree?

A] 10, 20, 15, 23, 25, 35, 42, 39, 30

B] 15, 10, 25, 23, 20, 42, 35, 39, 30

C] 15, 20, 10, 23, 25, 42, 35, 39, 30

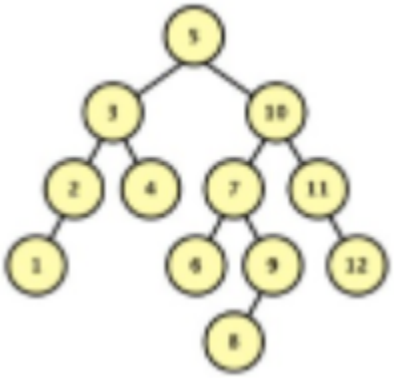
D] 15, 10, 23, 25, 20, 35, 42, 39, 30

Unit 2 AVL, Heap and TBT
Mapped course outcome: CO2

EASY

1. What is the time complexity of insertion, deletion, and access in a heap? Justify.
2. Explain with the help of example, how is a heap represented using an array? How do you find parent and child of a node in this representation.
3. Write the inorder traversal of a min heap of integers, after following operations: init, insert 30, insert 40, insert 20, insert 35, insert 22, remove, insert 25, remove, insert 8, insert 50
4. Consider a max heap, represented by the arrya: 40, 30, 20, 10,15,16,17,8,4. New element 35 is inserted in this heap. What would be the array contents after this inseretions?
5. A priority queue is implemented as a Max-Heap. Initially, it has 5 elements. The level-order traversal of the heap is: 10, 8, 5, 3, 2. Two new elements 1 and 7 are inserted into the heap in that order. The level-order traversal of the heap after the insertion of the elements is:
6. What is the maximum height of any AVL-tree with 7 nodes? Assume that the height of a tree with a single node is 0. Justify.
7. Write algorithm for LL rotation.
8. What are the rules for constructing the threads in a Threaded Binary Tree
9. In a Weight Balanced Tree - what is the meaning of weight? How is it calculated?
10. When is a tree called weight balanced? Give example of a weight balance tree.
11. Which all nodes will become imbalanced if a node is inserted as node g's child?

Medium

1. In a heap with n elements with the smallest element at the root, the 3th smallest element can be found in O(1) time. Explain.
2. Write code/algo for heapify() and adjust() functions which are called after insertion and deletion from a heap
3. Consider the process of inserting an element into a Max Heap, where the Max Heap is represented by an array. Suppose we perform a binary search on the path from the new leaf to the root to find the position for the newly inserted element, the number of comparisons performed is: θ(log2n) or θ(log2log2n) or θ(n) or θ(nlog2n)
4. Given the following AVL Tree:
5. 

   Draw the resulting BST after 5 is removed, but before any rebalancing takes place
   Label each node in the resulting tree with its balance factor
   Now rebalance the tree that results from (a). Draw a new tree for each rotation that occurs when rebalancing the
   AVL Tree (you only need to draw one tree that results from an RL or LR rotation). You do not need to label these trees with balance factors.
6. What are the number of rotations required to insert a sequence of elements 9, 6, 5, 8, 7, 10 into an empty AVL tree is?

7. What are weight-balanced trees? Explain with a suitable example
8. Give situations when each of LL, RL, RR and LR rotations will be invoked.
9. What are Threaded Binary Trees? List its advantages and applications.
10. Insert the folllowing nodes in a two way threaded tree. Show all steps.
11. 16 10 13 17 20 5 30 4 17
12. Consider the below left-left rotation pseudo code where the node contains value pointers to left, right child nodes and a height value and Height() function returns height value stored at a particular node.
    avltree leftrotation(avltreenode x):
    avltreenode w = x->left
    x->left = w->right
    w->right = x
    x- height = max(Height(x-left),
    Height(x->right))+1
    w->height = max(Blank 1)+1
    return w
    Fill missing code in Blank 1.
    What is the Time complexity of this code?
    Draw the part of tree before and after calling the function.
13. What are Height Balanced Tree. What is the motivation for such trees?
14. Create a BST and AVL Trees for numbers from 1 to 10. What is the difference in the height of two trees?
15. Why are AVL Trees called self balancing trees? What is its prime advantage?
16. How is Balance Factor calculated? Draw a tree with the numbers 45, 89, 62, 39, 44, 70, Write Balance Factor of each node.
17. Write a heap sort algorithm.

HARD

1. Write an algorithm to merge two heaps into one.
2. You are designing an application to display top k search queries on youtube. What is the most suitable data structure for this application? Why?
3. Write function to determine the type of imbalance in an AVL tree and returns an integer corresponding to each type (1,2,3 and 4). If there is no imbalance it should return 0.
4. Write an algorithm for inorder traversal of a Threaded Binary Tree
5. Why is Heap called a Priority Queue?

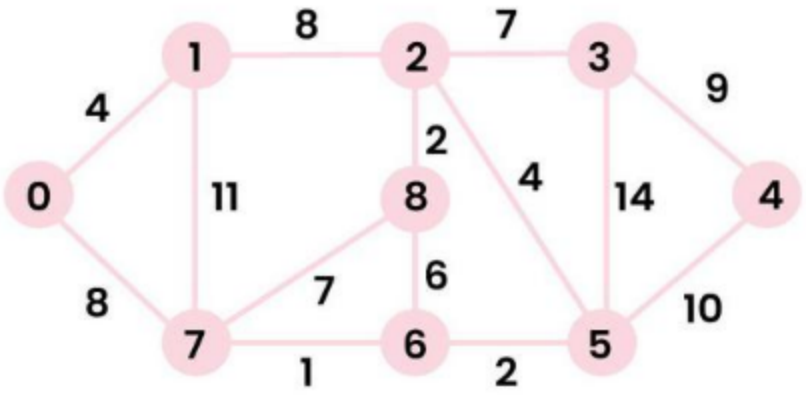
Unit 3: Graph
Mapped course outcome: CO3

Low level of difficulty questions :

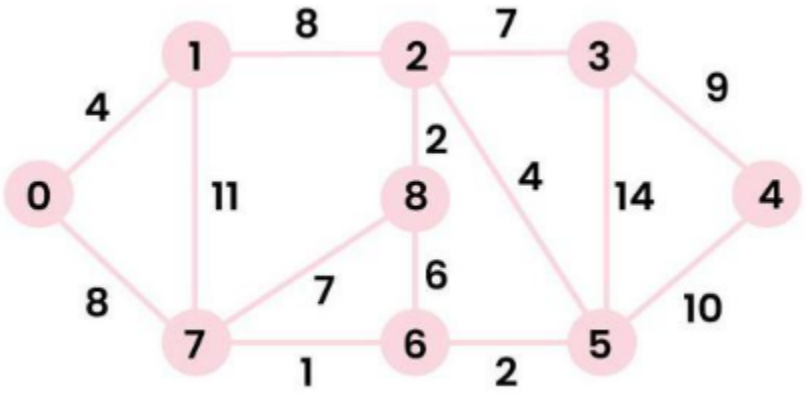
1. Differentiate 1) Weighted graph and unweighted graph breadth-first search (BFS) and depth-first search (DFS) Indegree and outdegree 4) directed and undirected graph
2. Illustrate the graph representation techniques with applications
3. State the difference between tree and graph with applications.
4. You're designing a map application. How would you represent cities and roads as a graph? Would you use an adjacency matrix or list, and why?
5. In a network of computers, how can BFS be used to find all devices reachable from a specific computer?
6. Write Dijkstra's algorithm with explanation
7. Write an algorithm for DFS traversal of a graph.
8. Write an algorithm for BFS traversal of a graph.

Medium level of difficulty questions :

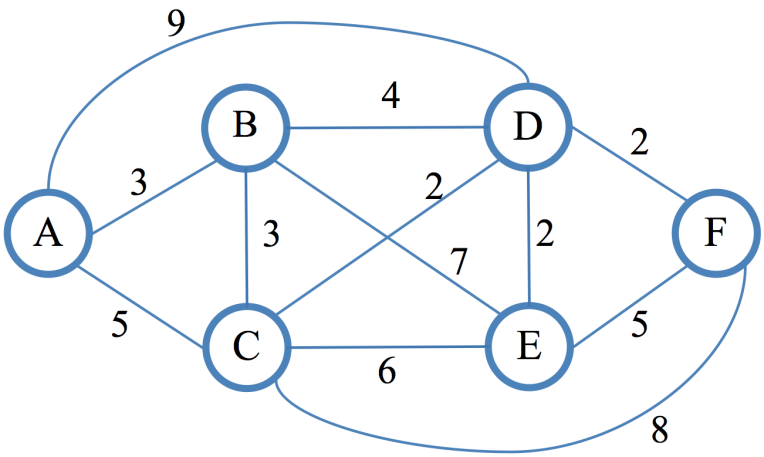
9. Write a function for DFS traversal of a graph.
10. Write a function for BFS traversal of a graph
11. Explore the suitability of DFS and BFS for finding the shortest path in an unweighted social network (number of "friend" hops).
12. Use DFS to find all reachable web pages from a given starting page on a website.
13. A company needs to lay fiber optic cables to connect multiple offices with minimal cost. Illustrate the working of Kruskal's algorithm for setting up the office.



14. Design a network by taking the suitable example (of 5 cities) of water pipes connecting several cities using Prim's algorithm to minimize the total pipe length.



15. Design a network by taking the suitable example (of 5 cities) of water pipes connecting several cities using Kruskal's algorithm to minimize the total pipe length.
16. Find the minimum cost to connect a group of islands with bridges using Prim's or Kruskal's algorithm.



17. Analyze the memory usage of an adjacency matrix vs. an adjacency list for a social network with millions of users and connections.
18. A web crawler needs to index all pages on a website. How would you use BFS or DFS to navigate the site's links?

    Consider a simplified website with the following page structure:

Web Page	Links To
Home (A)	About (B), Services (C)
About (B)	Contact (D), Blog (E)
Services (C)	Pricing (F), Blog (E)
Blog (E)	Contact (D)
Contact (D)	None
Pricing (F)	None

19. A project has dependencies between tasks. How can topological sorting be used to determine the order of tasks?
20. Use topological sorting to create a valid course schedule that satisfies all following courses and prerequisites:

Course	Prerequisite(s)
C1	None
C2	C1
C3	C1
C4	C2, C3
C5	C4
C6	C3

21. Design a system to find the shortest path between two people in a social network, considering friendship strength as edge weights.
22. Apply Dijkstra's algorithm to find the fastest route for a delivery truck, considering traffic conditions.
23. Compare adjacency matrix and adjacency list for representing road networks in a city.
24. In which real-life applications would an adjacency matrix be preferred over an adjacency list?
25. What is Dijkstra's algorithm, and how is it applied in GPS navigation?
26. How does Dijkstra's algorithm optimize airline route planning?

High level of difficulty questions :

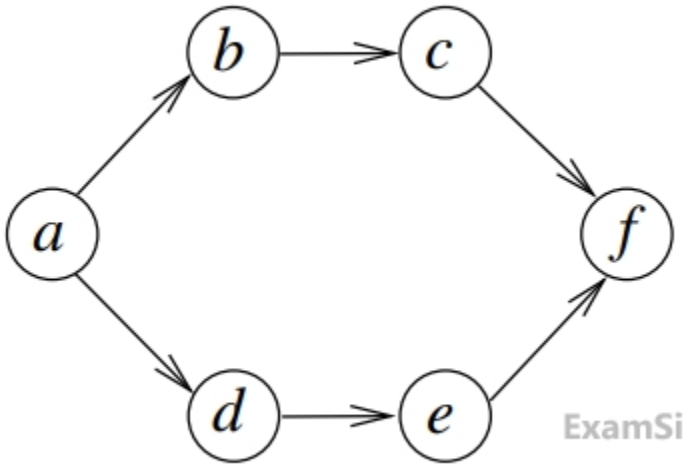
27. Evaluate the usefulness of topological sorting for resolving resource allocation conflicts in a complex project.
28. How can modifications to Dijkstra's algorithm make it useful for ride-sharing applications like Uber?
29. Analyze how a cycle in the dependency graph would affect the result of topological sorting.
30. Evaluate the efficiency of Prim's vs. Kruskal's algorithm for a very dense network of cities.

Gate questions:

31. In which scenario would a directed acyclic graph (DAG) be most suitable? Justify your answer.

1) Representing dependencies between tasks in a project schedule
2) Modeling a social network with friend connections
3) Finding the shortest path between two nodes in a weighted graph
4) Performing breadth-first search (BFS) on a graph

38. Consider the following directed graph:



39. Write the number of different topological orderings of the vertices of the graph with explanation.

40. Consider a complete undirected graph with a vertex set {0,1,2,3,4}. Entry Wij in the matrix W below is the weight of the edge {i, j}

$$W = \begin{pmatrix} 0 & 1 & 8 & 1 & 4 \\ 1 & 0 & 12 & 4 & 9 \\ 8 & 12 & 0 & 7 & 3 \\ 1 & 4 & 7 & 0 & 2 \\ 4 & 9 & 3 & 2 & 0 \end{pmatrix}$$

What is the minimum possible weight of a spanning tree T in this graph such that vertex 0 is a leaf node in the tree T?

41."A software development project consists of several modules (A, B, C, D, E, F, G) with dependencies between them. Each module requires specific resources (developers, testing environments, specialized software licenses) for its completion. The dependencies are as follows:

A must be completed before B and C.
B must be completed before D and E.
C must be completed before F.
D must be completed before G.
E must be completed before G.
F must be completed before G.

Furthermore, there are resource constraints. For instance:

Only 2 developers can work on modules concurrently.
The testing environment can only handle 1 module at a time.

Given these dependencies and resource constraints, determine a schedule that minimizes resource conflicts and ensures the project is completed efficiently."

Unit-4
Mapped course outcome: CO4

Low level of difficulty (6-8 Q) :

1. Illustrate indexing in the context of databases and data structures?
2. Compare and contrast B-Trees and AVL Trees in terms of indexing efficiency.

3. Discuss bitmap index, and where is it used?
4. State the key advantages of B-Trees in database indexing?
5. Discuss how does a B+ Tree differ from a B-Tree in terms of indexing?
6. “Self-balancing necessary for trees used in indexing”. Justify the statement.
7. Prove that how Multi-level Indexing optimize large database searches?
8. State the differences between Primary Index, Secondary Index, and Clustered Index.

Moderate level of difficulty (18-24 Q) :

1. Discuss the working of a B+ Tree with an example.
2. Discuss the process of insertion and deletion in B-Trees.
3. “Splay Tree achieve self-adjusting behavior”, illustrate with an example.
4. Show the rotations used in Red-Black Trees during insertion and deletion.
5. Compare the AA Tree and Red-Black Tree in terms of balancing and structure .
6. Explain the significance of AA Trees in terms of implementation simplicity.
7. Explain the significance of 2-3-4 Trees in relation to Red-Black Trees.
8. What is the role of leaf nodes in a B+ Tree?
9. “B-Tree handles underflow and overflow conditions”, Discuss in detail.
10. Prove that how does a B+ Tree improve range queries compared to a B-Tree?
11. “AVL Trees are not used in database indexing”, Justify the statement.
12. Discuss the cases of color flips and rotations in Red-Black Trees?
13. Enlist the practical applications of B-Trees in file systems and databases.
14. Discuss the properties of Red-black tree with example.
15. Illustrate AA tree and Red black tree with example.
16. Discuss how does Multi-level Indexing optimize large database searches?

Higher order thinking level (6-8 Q) :

1. Given the following set of keys: 10, 20, 5, 6, 30, 25, 40, construct a B-Tree of order 3
2. Given the following sequence: Insert: 15, 10, 20, 25, 30, 5, 35
   Construct A Red-Black Tree

3. Construct a B Tree for the following keys: 8, 16, 24, 32, 48, 56, 64 with order 4.
4. Explain the Insertion and Deletion operations in a Red-Black Tree with an example.
5. Given an initially empty Splay Tree, insert the keys 10, 20, 30, 40, 50 and show the final tree after each access operation.
6. Construct a B+ Tree of order 4 using the keys: 8, 16, 24, 32, 48, 56, 64, 72, 80 . Show all the intermediate steps.