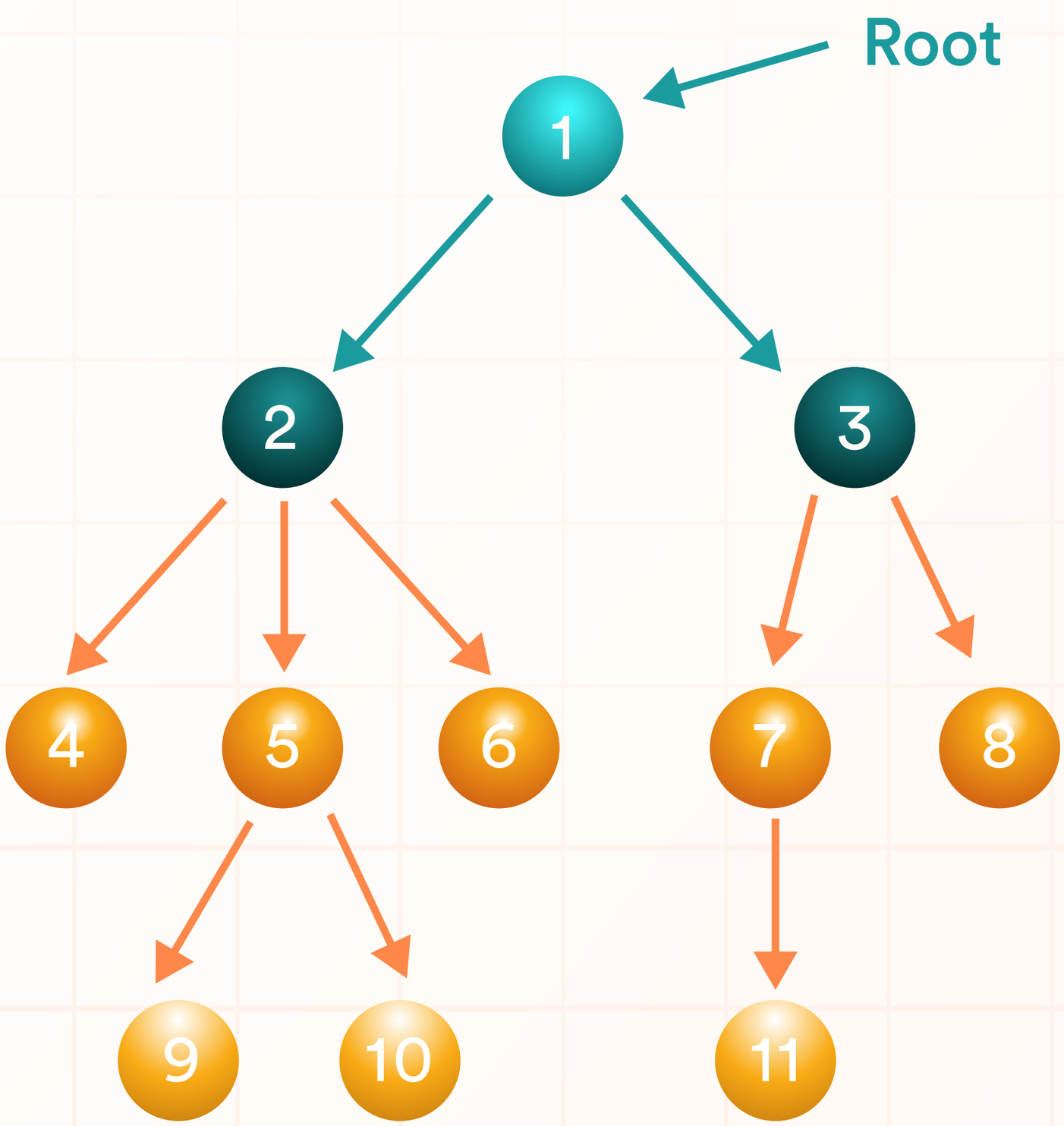
MASTER

TREES

IN 15 DAYS



DAY 1

Introduction to Trees

GOAL

Understand the basics of trees, become familiar with terminologies and tree properties.

TOPICS

- Introduction to trees and their properties
- Tree terminologies (root, node, leaf, etc.)
- Types of trees (binary, binary search, AVL, etc.)

RESOURCES

1. GeeksforGeeks - Trees: https://www.geeksforgeeks.org/tree-data-structure/

2. Introduction to Trees - Data Structure and Algorithms Tutorials: https://www.tutorialspoint.com/data_structures_algorithms/tree_data_structure.htm

3. Tree Data Structure - Codecademy: https://www.codecademy.com/learn/learn-trees

4. Tree (data structure) - Wikipedia: https://en.wikipedia.org/wiki/Tree_(data_structure)

Practice Questions

1. What is a tree in data structures?

2. Define the root of a tree.

3. What is a leaf node in a tree?

4. How are nodes connected in a tree?

5. What are the different types of trees?

6. Can a tree have multiple roots? Why or why not?

7. How is a binary tree different from other types of trees?

DAY 1

Binary Trees and Traversals

GOAL

Gain a solid understanding of binary trees and various traversal techniques.

TOPICS

- Binary tree representation and properties
- Pre-order, In-order, and Post-order traversals
- Level-order traversal (BFS)

RESOURCES

1. GeeksforGeeks - Binary Trees
https://www.geeksforgeeks.org/binary-tree-data-structure/

2. TutorialsPoint - Binary Tree Traversal
https://www.tutorialspoint.com/data_structures_algorithms/tree_traversal.htm

3. GeeksforGeeks - Level Order Traversal
https://www.geeksforgeeks.org/level-order-tree-traversal/

4. Binary Tree Traversal Visualization - VisuAlgo
https://visualgo.net/en/bst

Practice Questions

1. Perform an in-order traversal on the binary tree: 1 - 2 - 4 - 5 - 3.

2. Implement pre-order traversal iteratively without using recursion.

3. Perform a post-order traversal on the binary tree: 1 - 2 - 4 - 5 - 3.

4. Perform a level-order traversal (BFS) on the binary tree: 1 - 2 - 3 - 4 - 5.

5. Reconstruct the binary tree from the given pre-order traversal: 1 - 2 - 4 - 5 - 3.

6. Convert a binary search tree into a sorted doubly linked list using in-order traversal.

7. Check if the binary tree is a Full binary tree: 1 - 2 - 3 - 4 - 5.

DAY 3

Binary Search Trees (BST)

GOAL

Understand the concept of Binary Search Trees and their operations.

TOPICS

- Definition and properties of BST
- Insertion and deletion in BST
- Searching and finding the minimum/maximum element

RESOURCES

1. GeeksforGeeks - Binary Search Tree (BST):
https://www.geeksforgeeks.org/binary-search-tree-data-structure/

2. TutorialsPoint - Binary Search Trees:
https://www.tutorialspoint.com/data_structures_algorithms/binary_search_tree.htm

3. Programiz - Binary Search Tree (BST):
https://www.programiz.com/dsa/binary-search-tree

4. Data Structures | Binary Search Trees - Tutorialspoint:
https://www.studytonight.com/data-structures/binary-search-tree

Practice Questions

1. Write a function to insert a value into a Binary Search Tree (BST).

2. Implement a function to delete a node with a given value from a Binary Search Tree (BST).

3. Create a function to search for a specific value in a Binary Search Tree (BST).

4. Implement a function to find the minimum value in a Binary Search Tree (BST).

5. Write a program to find the maximum value in a Binary Search Tree (BST).

6. Create a function to check if a Binary Search Tree (BST) is valid, satisfying the BST properties.

7. Implement a function to perform an in-order traversal on a Binary Search Tree (BST) and print its elements.

DAY 4

Balanced Binary Search Trees (AVL Trees)

GOAL

Learn about AVL Trees and their self-balancing properties.

TOPICS

- Definition and properties of AVL Trees
- AVL Tree rotations (single and double rotations)
- Insertion and deletion in AVL Trees

RESOURCES

1. AVL Tree - GeeksforGeeks: https://www.geeksforgeeks.org/avl-tree-set-1-insertion/

2. AVL Trees | Data Structure Tutorial - Javatpoint: https://www.javatpoint.com/avl-tree

3. AVL Tree - Wikipedia: https://en.wikipedia.org/wiki/AVL_tree

4. AVL Trees - Topcoder: https://www.topcoder.com/thrive/articles/AVL%20Tree%20-%20Balancing%20Part%20I

Practice Questions

1. Write a function to insert a new node into an AVL Tree while maintaining its balance.

2. Implement a single left rotation operation for balancing an AVL Tree.

3. Perform a double rotation (left-right) on an AVL Tree to restore balance.

4. How would you handle the case of inserting duplicate values in an AVL Tree?

5. Given an AVL Tree, write a function to delete a specified node and rebalance the tree if necessary.

6. Implement a right rotation operation for balancing an AVL Tree.

7. Write a program to determine the height of an AVL Tree.

DAY 5

Binary Tree Traversal (Practice)

GOAL

Practice binary tree traversal algorithms through coding exercises.

TOPICS

- Pre-order, In-order, Post-order, and level-order traversal

RESOURCES

1. LeetCode: Tree Traversal Problems (Choose appropriate problems for practice)

Practice Questions

1. https://leetcode.com/problems/binary-tree-preorder-traversal/

2. https://leetcode.com/problems/binary-tree-inorder-traversal/

3. https://leetcode.com/problems/binary-tree-postorder-traversal/

4. https://leetcode.com/problems/binary-tree-level-order-traversal/

5. https://leetcode.com/problems/n-ary-tree-preorder-traversal/

6. https://leetcode.com/problems/n-ary-tree-postorder-traversal/

7. https://leetcode.com/problems/n-ary-tree-level-order-traversal/

8. https://leetcode.com/problems/binary-tree-zigzag-level-order-traversal/

9. https://leetcode.com/problems/populating-next-right-pointers-in-each-node/

10. https://leetcode.com/problems/binary-tree-right-side-view/

DAY 6

Binary Search Tree Operations (Practice)

GOAL

Strengthen your skills in BST operations through coding exercises.

TOPICS

- BST insertion, deletion, and searching
- Finding the minimum/maximum element in a BST

RESOURCES

LeetCode: Binary Search Tree Problems (Select appropriate problems for practice)

Practice Questions

1. https://leetcode.com/problems/insert-into-a-binary-search-tree/

2. https://leetcode.com/problems/delete-node-in-a-bst/

3. https://leetcode.com/problems/search-in-a-binary-search-tree/

4. https://leetcode.com/problems/minimum-distance-between-bst-nodes/

5. https://leetcode.com/problems/find-mode-in-binary-search-tree/

6. https://leetcode.com/problems/kth-smallest-element-in-a-bst/

7. https://leetcode.com/problems/increasing-order-search-tree/

8. https://leetcode.com/problems/binary-search-tree-to-greater-sum-tree/

9. https://leetcode.com/problems/balance-a-binary-search-tree/

10. https://leetcode.com/problems/construct-binary-search-tree-from-preorder-traversal/

DAY 7

AVL Tree Operations (Practice)

GOAL

Practice AVL Tree operations through coding exercises.

TOPICS

- AVL Tree insertion and deletion
- AVL Tree rotations

RESOURCES

GeeksforGeeks: AVL Tree Problems
https://www.geeksforgeeks.org/avl-tree-set-2-deletion/

Practice Questions

1. AVL Tree - Insertion: https://www.geeksforgeeks.org/avl-tree-set-1-insertion/

2. AVL Tree - Deletion: https://www.geeksforgeeks.org/avl-tree-set-2-deletion/

3. AVL Tree - Insertion and Rotations: https://www.geeksforgeeks.org/avl-tree-set-1-insertion/

4. AVL Tree - Insertion and Deletion Animation: https://www.geeksforgeeks.org/avl-tree-insertion-and-deletion/

5. AVL Tree - Check if it's Balanced: https://www.geeksforgeeks.org/check-if-a-given-binary-tree-is-avl-or-not/

6. AVL Tree - Convert to BST: https://www.geeksforgeeks.org/convert-normal-bst-balanced-bst/

7. AVL Tree - Maximum Height Difference: https://www.geeksforgeeks.org/find-maximum-height-difference-between-node-and-its-ancestor-in-binary-tree/

Practice Questions

8. AVL Tree - Inorder Traversal without Recursion: https://www.geeksforgeeks.org/inorder-tree-traversal-without-recursion/

9. AVL Tree - Sum of All Node Heights: https://www.geeksforgeeks.org/sum-of-all-the-nodes-having-the-right-child-as-a-child-node-in-an-avl-tree/

10. AVL Tree - Check if a Tree is a Subset of Another Tree: https://www.geeksforgeeks.org/check-if-a-binary-tree-is-subtree-of-another-binary-tree/

DAY 8

Binary Trees vs. Binary Search Trees

GOAL

Understand the similarities and differences between binary trees and binary search trees.

TOPICS

- Comparison of properties and characteristics
- Advantages and use cases of each

RESOURCES

GeeksforGeeks: Comparison between Binary Tree and Binary Search Tree
https://www.geeksforgeeks.org/difference-between-binary-tree-and-binary-search-tree/

Practice Questions

1. What is the main difference between a binary tree and a binary search tree?

2. Can a binary search tree have duplicate values? Why or why not?

3. In terms of searching for a specific value, which tree - binary tree or binary search tree - is more efficient and why?

4. What are the advantages of using a binary search tree over a regular binary tree?

5. Describe a scenario where a binary search tree would be a better choice than a binary tree for storing and retrieving data.

6. Explain how the insertion of elements differs in a binary tree and a binary search tree.

DAY 9

Priority Queues and Heaps

GOAL

Learn about priority queues and heap data structure.

TOPICS

- Definition and properties of priority queues
- Types of heaps (min-heap and max-heap)
- Operations on heaps (insertion, extraction, heapify)

RESOURCES

1. GeeksforGeeks - Priority Queues: https://www.geeksforgeeks.org/priority-queue-set-1-introduction/

2. Data Structures and Algorithms - Heaps and Priority Queues: https://www.coursera.org/lecture/data-structures/heap-and-priority-queue-3o5dw

3. Introduction to Heap Data Structure - Tutorialspoint: https://www.tutorialspoint.com/data_structures_algorithms/heap_data_structure.htm

4. Heap Data Structure - Stanford CS Education Library: https://cslibrary.stanford.edu/110/BinaryHeaps.html

Practice Questions

1. What is the main purpose of a priority queue, and what properties does it have?

2. What are the two main types of heaps, and how do they differ from each other?

3. What is the process of "heapify" in a heap data structure?

4. In a min-heap, which child does a parent node compare itself with during heapify?

5. Implement a priority queue using an array-based approach with functions for insertion and extraction.

6. Given an array of integers, use a min-heap to find the Kth smallest element efficiently.

7. Write a function to perform heapify on an array representing a binary tree at a given

8. Implement a max-heap data structure with a function to update the priority of a specific element.

9. Use a min-heap to merge K sorted arrays into a single sorted array efficiently.

DAY 10

Heap Sort

GOAL

Learn how to implement the Heap Sort algorithm.

TOPICS

- Overview of Heap Sort algorithm
- Steps involved in Heap Sort
- Time and space complexity analysis

RESOURCES

1. GeeksforGeeks - Heap Sort:
https://www.geeksforgeeks.org/heap-sort/

2. Tutorialspoint - Heap Sort:
https://www.tutorialspoint.com/data_structures_algorithms/heap_sort_algorithm.htm

3. Khan Academy - Heap Sort:
https://www.khanacademy.org/computing/computer-science/algorithms/heap-sort/a/heap-sort

Practice Questions

1. Explain the steps involved in the Heap Sort algorithm.

2. What are the time and space complexities of Heap Sort?

3. Implement Heap Sort and apply it to a sample array.

4. Implement the Heap Sort algorithm to sort an array of integers in ascending order.

5. Explain the steps involved in building a max heap from an array of integers. Write a function to demonstrate this process.

6. Write a function to find the kth smallest element in an unsorted array using the Heap Sort algorithm.

7. Design a function to determine if a given binary tree is a max heap or not.

DAY 11

Tries

GOAL

Learn about trie data structure and its applications.

TOPICS

- Definition and properties of trie
- Trie operations (insertion, deletion, searching)
- Use cases of trie (dictionary, autocomplete)

RESOURCES

1.GeeksforGeeks: Trie Data Structure
https://www.geeksforgeeks.org/trie-insert-and-search/

2. TutorialsPoint:
https://www.tutorialspoint.com/data_structures_algorithms/trie_data_structure.htm

3. https://leetcode.com/articles/implement-trie-prefix-tree/

Practice Questions

1. What is a trie, and why is it used for efficient string searching?

2. Implement algorithms for trie insertion, deletion, and searching.

3. How can you use a trie for implementing a dictionary or autocomplete feature?

4. Implement a trie data structure that supports insertion and searching of words. Write functions to insert words into the trie and check if a given word exists in the trie.

5. Given an array of strings, construct a trie to represent the dictionary. Then, implement a function that takes a prefix as input and returns all words from the dictionary that start with that prefix.

6. Implement a function to delete a word from the trie. If the deleted word has a common prefix with other words in the trie, the remaining characters should be preserved as part of those words.

7. Given a list of strings, find the longest common prefix among them using a trie data structure. Return an empty string if there is no common prefix.

DAY 12-15

Tree Applications and Practice

GOAL

Explore real-world applications of trees and practice various tree-related problems.

TOPICS

- Huffman Coding and tree-based compression techniques
- Segment Trees for range queries
- Fenwick (Binary Indexed) Trees

RESOURCES

1.GeeksforGeeks: Tree-based Problems and Applications (Choose appropriate problems)

2.LeetCode: Tree Problems (Select appropriate problems for practice)

Practice Questions

1. GeeksforGeeks: Huffman Decoding: https://www.geeksforgeeks.org/huffman-decoding/

2. LeetCode: Serialize and Deserialize Binary Tree: https://leetcode.com/problems/serialize-and-deserialize-binary-tree/

3. GeeksforGeeks: Huffman Encoding: https://www.geeksforgeeks.org/huffman-coding-greedy-algo-3/

4. LeetCode: Encode and Decode Strings: https://leetcode.com/problems/encode-and-decode-strings/

5. GeeksforGeeks: Segment Tree | Set 1 (Sum of given range): https://www.geeksforgeeks.org/segment-tree-set-1-sum-of-given-range/

6. LeetCode: Range Sum Query - Mutable: https://leetcode.com/problems/range-sum-query-mutable/

7. GeeksforGeeks: Binary Indexed Tree or Fenwick Tree: https://www.geeksforgeeks.org/binary-indexed-tree-or-fenwick-tree-2/

Practice Questions

8. LeetCode: Range Sum Query 2D - Mutable: https://leetcode.com/problems/range-sum-query-2d-mutable/

9. LeetCode: Binary Tree Level Order Traversal: https://leetcode.com/problems/binary-tree-level-order-traversal/

10. GeeksforGeeks: Inorder Successor in BST: https://www.geeksforgeeks.org/inorder-successor-in-binary-search-tree/

BOSSCODER ACADEMY

WHY BOSSCODER?

750+ Alumni placed at Top Product-based companies.

More than **136% hike** for every **2 out of 3** working professional.

Average package of **24LPA.**

The syllabus is most **up-to-date** and the list of problems provided covers **all important topics.**

Lavanya
Meta

Course is very well **structured** and **streamlined** to crack any **MAANG** company

Rahul .
Google

EXPLORE MORE