Organized Data Structures Topics

Binary Tree

Theory Questions

- What is the depth of a node?
- What is the height of a node vs depth of a node?
- What is a complete binary tree?
- What is a full binary tree?
- What is a perfect binary tree?
- What is a degenerate binary tree?
- What is a balanced vs unbalanced binary tree?
- What are internal nodes?
- What are siblings?
- What is the degree of a node?
- What are the terminologies of a tree?
- What are the applications of trees?
- What is the difference between a binary tree and a binary search tree (BT vs BST)?
- What is an N-ary tree?
- What is a ternary tree?
- What is a segment tree?
- What is the time complexity to search in a binary tree?
- What is the difference between a complete tree vs full tree vs perfect tree?
- What is a linear vs non-linear data structure?
- What is the difference between a tree vs graph?
- What is the lowest common ancestor (LCA) in a binary tree?
- What is a subtree?
- What is the time complexity of level order traversal?
- What is the time complexity of preorder, postorder, and inorder traversal?
- What are the data structures used for DFS and BFS traversal of a binary tree?

Practical Questions

- Create a binary tree and calculate its height.
- Implement a binary tree (not BST) and show how the insert function works.
- Implement a tree structure with multiple children per node and display its elements.
- Write a function to check if a tree is balanced or not.

- Perform level order traversal of a binary tree.
- Implement preorder traversal.
- Implement postorder traversal.
- Implement DFS and BFS in a binary tree.
- Check if two binary trees are identical.
- Check if one binary tree is a subtree of another.
- Print all leaf nodes in a binary tree.
- Find the lowest common ancestor (LCA) in a binary tree.

Binary Search Tree (BST)

Theory Questions

- What is a binary search tree (BST)?
- What are the applications of BST?
- What is the difference between BST vs binary tree?
- What is the time complexity of BST insertion?
- What is the time complexity of searching in a BST?
- What is the time complexity of removing the second largest element in a BST?
- What is the difference between BST vs max heap?
- What are the advantages of allowing duplicate elements in a BST?
- What is the complexity of BST operations (search, insert, delete)?
- What is the degree of node vs degree of tree in a BST?
- What is a degenerate tree in the context of BST?
- How to validate whether a given tree is a BST or not?
- What is the second largest element in a BST?
- What is the third largest element in a BST?
- What is the kth smallest element in a BST?
- What is the kth largest element in an array using BST?
- How to count single child nodes in a BST?
- What are the special cases in BST deletion?

Practical Questions

- Create a Binary Search Tree and perform traversal (inorder, preorder, postorder).
- Implement a BST.
- Find the height of a BST.
- Validate whether a given tree is a BST or not (handle negative scenarios).
- Find the second largest element in a BST.
- Find the third largest element in a BST.
- Find the kth smallest element in a BST (use inorder traversal).
- Find the kth largest element in an array using BST.
- Find the element closest to a target value in a BST.

- Allow duplicate elements in a BST.
- Implement BST deletion (including special cases).
- Count single child nodes in a BST.
- Find the minimum value in a BST using recursion.
- Check if a BST is balanced.

AVL Tree

Theory Questions

- What is an AVL tree?
- What is a balanced BST?
- What are rotations in AVL trees?
- What are the applications of AVL trees?
- What is the time complexity of operations in an AVL tree?

Practical Questions

- Implement an AVL tree.
- Perform rotations in an AVL tree to balance it.

Red-Black Tree

Theory Questions

- What is a red-black tree?
- What are the properties of a red-black tree?
- What are the applications of red-black trees?
- What is the time complexity of operations in a red-black tree?

Practical Questions

- Implement a red-black tree.
- Find the second largest element in a red-black tree.
- Find the third largest element in a red-black tree.

Heap

Theory Questions

- What is a heap?
- What is a max heap vs min heap?

- What is the max-heap and min-heap property?
- What are the applications of heaps?
- What is the time complexity of heap sort?
- What is the time complexity of inserting a value in a heap?
- What is the time complexity of deleting a node from a heap?
- What is heapify (heapify up and heapify down)?
- What is the limitation of a heap?
- Is a heap a complete binary tree?
- What is the difference between BST vs max heap?
- What is the application of a priority queue?
- What is the time complexity of converting a min heap to a max heap?
- How to find the right child of a heap?
- What is the time complexity of initializing a heap?
- What is the purpose of heap sort?
- What is the complexity of heapify?

Practical Questions

- Create a max heap, insert elements, and display.
- Implement a min heap.
- Implement heap sort.
- Implement a priority queue using a heap.
- Convert a min heap to a max heap.
- Delete a node from a min heap.
- Delete a node from a max heap.
- Find the kth largest element in an array using a heap.
- Find the top k frequent elements using a heap.
- Implement heapify up and down.
- Build a heap from an array.

Trie

Theory Questions

- What is a trie (prefix tree)?
- What are the types of tries?
- What is a suffix trie vs prefix trie?
- What is a compressed trie?
- What are the applications of tries?
- What are the advantages of tries?
- What is the complexity of initializing a trie?
- What is the complexity of trie operations (insert, search, delete)?
- What is auto-completion using a trie?

- What is prefix search in a trie?
- What is the longest prefix in a trie?
- What is trie serialization and deserialization?
- What is a self-balancing trie?
- What is a radix tree?

Practical Questions

- Implement a trie .
- Implement auto-completion using a trie.
- Implement prefix search using a trie.
- Find the longest prefix in a trie.
- Implement trie serialization and deserialization.
- Insert a new word into a trie.
- Search a word in a trie.
- Delete a word from a trie.
- Find the longest non-repeating substring in a string using a trie.

Graph

Theory Questions

- What is a graph?
- What is the difference between a directed vs undirected graph?
- What is a weighted graph vs unweighted graph?
- What is a complete graph?
- What is a disconnected graph?
- What is a bipartite graph?
- What is the degree of a vertex?
- What is the degree of a node vs degree of a tree?
- What are the types of graphs?
- What are the applications of graphs?
- What is a minimum spanning tree?
- What is the use of a spanning tree?
- What is the complexity of BFS in a graph?
- What is the complexity of DFS in a graph?
- What is the complexity of initializing a graph?
- What are the ways to implement a graph?
- What is an adjacency matrix vs adjacency list?
- What is the difference between a tree vs graph?
- How are graphs used in social media to find mutual friends?
- How to detect cycles in a graph?
- How to count cycles in a graph?

- What is the shortest path in a graph?
- What is Dijkstra's algorithm for shortest path in a weighted graph?
- What is the shortest path in an unweighted graph using BFS?
- What is backtracking in DFS?
- What is a loop in a graph?
- What are Prim's and Kruskal's algorithms?
- What is the time complexity of graph operations?
- What is graph indexing?
- How to represent a graph in memory?
- What is the classification of graphs?

Practical Questions

- Implement a graph with traversal using DFS.
- Implement BFS in a graph.
- Clone a graph (LeetCode 133).
- Find the shortest path in an unweighted graph using BFS.
- Implement Dijkstra's algorithm for shortest path in a weighted graph.
- Detect cycles in a graph.
- Count cycles in a graph.
- Find the shortest distance between two vertices in a graph.
- Implement a minimum spanning tree using Prim's algorithm.
- Implement a minimum spanning tree using Kruskal's algorithm.
- Solve LeetCode 200: Number of Islands.
- Implement graph traversal (DFS and BFS).
- Check if a graph is bipartite.
- Find mutual friends in a social media graph.

Hashing

Theory Questions

- Why should a hashtable be used when an array has similar time complexity?
- What are popular hashing algorithms?
- What is double hashing?
- What are the applications of hash tables?
- What is the time complexity of hash table operations?

Practical Questions

- Implement a hash table.
- Sort transaction objects based on the `.amount` property.
- Implement double hashing.

Additional Topics

Theory Questions

- What is the shortest path in a graph?
- What is the complexity of various algorithms?
- What is quadratic time complexity?
- What is linear time complexity?
- What is the advantage of recursion?
- What is the sliding window pattern?
- What is the complexity of all data structures?
- What is Big O notation?
- What are logarithmic values and functions?

Practical Questions

- Implement a queue with the ability to dequeue from both ends (edge cases not handled).
- Find unique characters from a given string.
- Find a combination of numbers that sum to a target value (e.g., sum = 2 from `[1,2,-3,5,0,4,-8,-5,7,-9,-7,3]`).
- Solve problems from Blind 75 LeetCode and learn optimal solutions by watching YouTube.
- Solve LeetCode 102: Binary Tree Level Order Traversal.
- Solve LeetCode 133: Clone Graph.
- Solve LeetCode 230: Kth Smallest Element in a BST.
- Solve LeetCode 215: Kth Largest Element in an Array.
- Solve LeetCode 200: Number of Islands.
- Debug and fix issues in code.
- Increase coding speed and problem-solving skills (complete complex problems within 5 minutes).
- Practice more logical workouts on platforms like LeetCode, HackerRank, and GeeksforGeeks.