Sudarshan Yadav

Q1: What is a Data structure?

Ans: **Data structure** is a way of **organizing and storing data** in a computer so that it can be used efficiently.

Q2: What is an Algorithm?

Ans: **Algorithm** is a **step-by-step procedure or a set of rules for solving specific problem or accomplishing specific task.**

Q3: What is the difference between an Array and a Linked list?

Ans: **Array** is a **collection of elements of the same data type, stored in contiguous memory locations**, while

Linked list is a **collection of elements called nodes, where each node contains both data & reference to the next node.**

Q4: What is a Linked list?

Ans: **Linked list** is a **data structure consisting of** a **sequence of nodes, where each node contains data and a reference to the next node in the sequence.**

Q5: What is a Stack?

Ans: **Stack** is a **linear data structure** that follows the **Last-In-First-Out (LIFO) principle**, where the **last element inserted is the first one to be removed.**

Q6: What is a Queue?

Ans: **Queue** is a **linear data structure** that follows the **First-In-First-Out (FIFO) principle**, **where the first element inserted is the first one to be removed.**

Q7: What is the Time complexity of inserting an element at the end of an array?

Ans: **Time complexity of inserting an element at the end of an array** is **O(1).**

Q8: What is the Time complexity of binary search?

Ans: **Time complexity of binary search** is **O(log n), where n is the number of elements in the sorted array.**

Q9: What is the Time complexity of linear search?

Ans: **Time complexity of linear search** is **O(n), where n is the number of elements in the array.**

Q10: What is the Time complexity of inserting an element at the beginning of a linked list?

Ans: **Time complexity of inserting an element at the beginning of a linked list is O(1).**

Q11: What is the Time complexity of bubble sort?

Ans: **Time complexity of bubble sort is $O(n^2)$, where n is the number of elements in the array.**

Q12: What is the Time complexity of depth-first search (DFS)?

Ans: **Time complexity of depth-first search (DFS) is O(V + E), where V is the number of vertices and E is the number of edges in the graph.**

Q13: What is the Time complexity of selection sort?

Ans: **Time complexity of selection sort is $O(n^2)$, where n is the number of elements in the array.**

Q14: What is the Time complexity of insertion sort?

Ans: **Time complexity of insertion sort is $O(n^2)$, where n is the number of elements in the array.**

Q15: What is the Time complexity of merge sort?

Ans: **Time complexity of merge sort is O(n log n), where n is the number of elements in the array.**

Q16: What is the Time complexity of quicksort?

Ans: **Time complexity of quicksort is O(n log n), where n is the number of elements in the array.**

Q17: What is the Time complexity of searching in a binary search tree (BST)?

Ans: **Time complexity of searching in a binary search tree (BST)** is **O(log n), where n is number of nodes in the tree.**

Q18: What is the Time complexity of heap sort?

Ans: **Time complexity of heap sort is O(n log n), where n is the number of elements in the array.**

Q19: What is the Time complexity of breadth-first search (BFS)?

Ans: **Time complexity of breadth-first search (BFS) is O(V + E), where V is the number of vertices and E is the number of edges in the graph.**

Q20: What is the time complexity of the AVL tree operations?

Ans: **Time complexity of AVL tree operations (insertion, deletion, and search) is O(log n), where n is the number of nodes in the tree.**

Q21: What is a Hash table?

Ans: **Hash table** is a **data structure that implements an associative array abstract data type**, **where data is stored in an array based on its key.**

Q22: What is a Hash function?

Ans: **Hash function** is a **function that takes an input (or key) and returns a fixed-size numerical value called a hash code or hash value.**

Q23: What is a Binary tree?

Ans: **Binary tree** is a **tree data structure** in which **each node has at most two children, referred to as the left child and the right child.**

Q24: What is a Graph in data structures?

Ans: **Graph** is a **non-linear data structure** consisting of a **set of vertices (nodes) and a set of edges (connections) between the vertices.**

Q25: What is a Graph traversal?

Ans: **Graph traversal** is the **process of visiting all the nodes of a graph in a specific order.**

Q26: What is a Binary search tree (BST)?

Ans: **Binary search tree** is a **binary tree in which for each node,** the **value of all the nodes in its left subtree is less than its value,** and the **value of all the nodes in its right subtree is greater than its value.**

Q27: What is a Trie?

Ans: **Trie**, also known as a **prefix tree**, is a **tree-like data structure** that **stores a collection of strings and provides efficient prefix-based search operations.**

Q28: What is a Priority queue?

Ans: **Priority queue** is a **data structure that stores elements along with their associated priorities and allows retrieval of the element with the highest priority.**

Q29: What is a Balanced binary search tree?

Ans: **Balanced binary search tree** is a **binary search tree** that **maintains a balance condition**, **such as AVL tree or Red-Black tree**, to ensure **efficient search**, **insertion,** and **deletion operations**.

Q30: What is Dynamic programming?

Ans: **Dynamic programming** is a **technique used to solve complex problems by breaking them down into simpler overlapping subproblems and solving each subproblem only once.**

Q31: What is a Disjoint set data structure?

Ans: **Disjoint set data structure** is a **data structure** that **keeps track of a partitioning of a set into disjoint subsets and provides efficient operations to merge sets and find the representative element of a set.**

Q32: What is a Heap data structure?

Ans: **Heap** is a **complete binary tree** that **satisfies the heap property**, which can be either a **min-heap (where the parent node is smaller than or equal to its children)** or a **max-heap (where the parent node is larger than or equal to its children).**

Q33: What is a Suffix array?

Ans: **Suffix array** is a **sorted array of all suffixes of a given string**, **used in string processing algorithms such as pattern matching and substring search.**

Q34: What is a B-tree?

Ans: **B-tree** is a **self-balancing search tree data structure** that **maintains sorted data and allows efficient operations such** as **insertion**, **deletion**, and **search**.

Q35: What is dynamic memory allocation?

Ans: **Dynamic memory allocation** is the **process of allocating and deallocating memory at runtime, allowing programs to manage memory dynamically as per their requirements.**

Q36: What is a Hash collision?

Ans: **Hash collision** occurs when **two different keys produce the same hash value**, and **Hash table needs to resolve this conflict to store both keys.**

Q37: What is the time complexity of the A* algorithm for finding the shortest path in a graph with heuristic estimates?

Ans: **Time complexity of the A* algorithm** depends on the **specific problem and the heuristic function used**. **It can range from linear time to exponential time.**

Q38: What is memoization?

Ans: **Memoization** is an **optimization technique used in dynamic programming where the results of expensive function calls are stored and reused to avoid redundant computations.**

Q39: What is a self-balancing binary search tree?

Ans: **Self-balancing binary search tree** is a **binary search tree** that automatically **maintains a balanced structure during insertions and deletions to ensure efficient search**, **insertion**, and deletion operations.

Q40: What is a Tree in data structures?

Ans: **Tree** is a **nonlinear data structure composed of nodes connected by edges**, **where each node can have zero or more children.**