Data Structure & Algorithm Viva

1. **What is data structure? (imp)**
   Ans:- Data structure refers to the way data is organized and manipulated. It seeks to find ways to make data access more efficient. When dealing with the data structure, we not only focus on one piece of data but the different set of data and how they can relate to one another in an organized manner.

2. **Differentiate between file and structure storage structure. (imp)**
   Ans:-The key difference between both the data structure is the memory area that is being accessed. When dealing with the structure that resides the main memory of the computer system, this is referred to as storage structure. When dealing with an auxiliary structure, we refer to it as file structures.

3. **What is an ordered list? (imp)**
   Ans:- an ordered list refers to a linear data structure where elements are arranged in a specific order. The order could be based on numerical values, alphabetical order, or any other defined criteria.

4. **What is a linked list? (imp)**
   Ans:-A linked list is a sequence of nodes in which each node is connected to the node following it. This forms a chain-like link for data storage.
5. **What are doubly linked lists? (imp)**
   Ans:- A doubly linked list is a type of linked list in which each node contains a data element and two links, one pointing to the next node and another pointing to the previous node. This allows traversal in both forward and backward directions.

6. **What is LIFO? (imp)**
   Ans:-LIFO is a short form of Last In First Out. It refers how data is accessed, stored and retrieved. Using this scheme, data that was stored last should be the one to be extracted first.

7. **What is FIFO? (imp)**
   Ans:-FIFO stands for First-in, First-out, and is used to represent how data is accessed in a queue. In a FIFO structure, the first element added to the structure is the first one to be removed.

8. **What is a stack? (imp)**
   Ans:-A stack is a data structure in which only the top element can be accessed. In a stack, the last element added is the first one to be removed..

9. **What is a queue? (imp)**
   Ans:-A queue is a data structure that can simulate a list or stream of data. In this structure, new elements are inserted at one end, and existing elements are removed from the other end. Opration of queue 1. Dequeue 2. enqueue
10. **What is a dequeue And enqueue ?**
    Ans:- Dequeue is the operation of removing an element from the front of a queue. It is used to retrieve and remove the element from the front of the queue. While Enqueue is the operation of adding an element to the rear (end) of a queue. It is used to insert a new element into the queue.

11. **What are binary trees? (imp)**
    Ans:-A binary tree is one type of data structure that has two nodes, a left node, and a right node. In programming, binary trees are an extension of the linked list structures.
12. **Explain Binary Search Tree ? (imp)**
    Ans:-A binary search tree stores data in such a way that they can be retrieved very efficiently. The left subtree contains nodes whose keys are less than the nodes key value, while the right subtree contains nodes whose keys are greater than or equal to the nodes key value. Moreover, both subtrees are also binary search trees.

13. **What is a linear search? (imp)**

Ans:- A linear search refers to the way a target key is being searched in a sequential data structure. In this method, each element in the list is checked and compared against the target key. The process is repeated until found or if the end of the file has been reached.

14. **What is Binary search ?**

Ans:- Binary search is a searching algorithm used to find the position of a target value within a sorted array or list. It is a highly efficient algorithm with a time complexity of O(log n), where 'n' is the number of elements in the array.

15. **Differentiate STACK from ARRAY. (imp)**

Ans:- Stack follows a LIFO pattern. It means that data access follows a sequence wherein the last data to be stored when the first one to be extracted. Arrays, on the other hand, does not follow a particular order and instead can be accessed by referring to the indexed element within the array.

16. **Types of sorting ?**

Sorting is the process of arranging elements in a particular order, usually in ascending or descending order, based on certain criteria.

Ans:- 1. **Bubble Sort** **2. Selection Sort**
3. Merge Sort **4. Quick Sort**

17. **What is a bubble sort and how do you perform it? (imp)**

Ans:- A bubble sort is one sorting technique that can be applied to data structures such as an array. It works by comparing adjacent elements and exchanges their values if they are out of order.

18. **What is a Selection Sort ?**

Ans:- It repeatedly finds the minimum element from the unsorted part of the array and puts it at the beginning.

19. **What is merge sort? (imp)**

Ans:- Merge Sort is a popular sorting algorithm that follows the divide-and-conquer paradigm. It divides the input array into two halves, recursively sorts each half, and then merges the sorted halves to produce a fully sorted array.

20. **What is a Quick Sort ?**

Ans:- It picks a 'pivot' element and partitions the array into two sub-arrays, then recursively sorts each sub-array.

21. **Differentiate NULL and VOID (imp)**

Ans:- Null is a value that indicates no value or no valid memory address. used in pointers to signify that the pointer is not pointing to any valid memory location. While VOID is a data type that represents the absence of a value. commonly used in functions to indicate that the function does not return any value.

22. **What is the difference between a PUSH and a POP? (imp)**

Ans:- PUSH is an operation that adds an element to the top of a stack It is used to insert a new element onto the stack. While POP is an operation that removes the top element from a stack. It is used to retrieve and remove the element from the top of the stack.

23. **What is a postfix expression?**

Ans:- A postfix expression is an expression in which each operator follows its operands. The advantage of this form is that there is no need to group sub-expressions in parentheses or to consider operator precedence.

24. **What are ARRAYS? (imp)**

Ans:- an array like a row of boxes where we can store different items. Each box has a number, starting from the first box at 0, then 1, 2, and so on. we can easily put things in these boxes or take them out based on their numbers.

25. **Which one is fast sorting algorithm explain ?**
Ans:- QuickSort is considered a fast sorting algorithm, especially for larger datasets, when compared to BubbleSort.

26. **What is Recursion ?**
Ans:- Recursion is a programming concept where a function calls itself in its own definition. In simpler terms, it's like a process that repeats itself in a self-similar way. Recursion is often used when a problem can be broken down into smaller subproblems of the same type.

27. **What is depth-First search ?**
Ans:- Depth-First Search (DFS) is a graph traversal algorithm that explores as far as possible along each branch before backtracking. It goes deep into the structure before exploring other branches. It's often used to explore and analyze graphs.

28. **What is Breadth-First Search ?**
Ans:- Breadth-First Search (BFS) is a graph traversal algorithm that explores all the vertices of a graph level by level. BFS is often used to find the shortest path in an unweighted graph.

29. **What is a Tree ?**
Ans:- A tree is a hierarchical data structure that consists of nodes connected by edges. It is widely used in computer science for organizing and structuring data. A tree has a root node, and each node has zero or more child nodes. The node without any parent is called the root, and nodes with no children are called leaves.

30. **What is Graph ?**
Ans:- A graph is a non-linear data structure that consists of a collection of nodes (vertices) and edges connecting pairs of nodes. The edges may have a direction (directed graph) or no direction (undirected graph).

31. **What is Time Complexity?**
Ans:- Time complexity is a measure of the amount of time an algorithm takes to complete based on the size of the input. It is expressed using big O notation (O(f(n))), where "f(n)" represents the dominant term that contributes the most to the overall growth of the algorithm as the input size increases.

32. **What is Space complexity ?**
Ans:- Space complexity is a measure of the amount of memory an algorithm uses based on the size of the input. It also uses big O notation (O(f(n))), where "f(n)" represents the dominant term that contributes the most to the overall growth of memory usage as the input size increases.

33. **Function in DSA ?**
Ans:- Functions in Data Structures:
Enqueue: A function used in queues to add an element to the rear of the queue**.**
Dequeue: A function used in queues to remove an element from the front of the queue.
Insert: A function used in linked lists to add a new node at a specified position.
Delete: A function used in linked lists to remove a node from a specified position.
Search: A function used to find the presence of an element in a data structure.

34. **What is Real and Front queue ?**
Ans;- The "Rear" or "Real" refers to the last element in the queue, which is the one that was enqueued most recently. While "Front" refers to the front or first element in the queue.

35. **What is Fibonacci search? (imp)**

Ans:- Fibonacci search is a search algorithm that applies to a sorted array. It makes use of a divide- and-conquer approach that can significantly reduce the time needed in order to reach the target element.

36. **What is Data abstraction? (imp)**

Ans:- Data abstraction is a powerful tool for breaking down complex data problems into manageable chunks. This is applied by initially specifying the data objects involved and the operations to be performed on these data objects without being overly concerned with how the data objects will be represented and stored in memory.

37. **What are dynamic data structures? (imp)**

Ans:- Dynamic data structures are structures that expand and contract as a program runs. It provides a flexible means of manipulating data because it can adjust according to the size of the data.

38. **Briefly explain recursive algorithm. (imp)**

Ans:- Recursive algorithm targets a problem by dividing it into smaller, manageable sub-problems. The output of one recursion after processing one sub-problem becomes the input to the next recursive process.

39. **What is Huffmans algorithm? (imp)**

Ans:- Huffmans algorithm is used for creating extended binary trees that have minimum weighted path lengths from the given weights. It makes use of a table that contains the frequency of occurrence for each data element

40. **What is Notation ?**

Ans:- "notation" typically refers to the analysis and representation of the performance or efficiency of algorithms. There are two primary notations used in DSA:

1. **Big O Notation (O):** Big O notation describes the upper bound or worst-case complexity of an algorithm.
2. **Theta Notation (Θ):** Theta notation provides a tight bound on the growth rate of an algorithm, representing both the upper and lower bounds.

41. **What is significant of time complixity in programming?**

Ans:- The significance of time complexity in programming lies in understanding and analyzing the efficiency of algorithms. Time complexity is a measure of the amount of time an algorithm takes to complete as a function of the size of the input.

42. **What is Syntax Error ?**

Ans:- A syntax error is a type of error that occurs in a program when there is a violation of the programming language's syntax rules.

43. **What is Dynamic Memory allocation ?**

Ans:- Dynamic memory allocation refers to the process of allocating memory for variables at runtime (during program execution) rather than at compile time.

44. **What is a algorithm?**

Ans:- An algorithm is a step-by-step procedure or set of rules designed to perform a specific task or solve a particular problem. It is a finite sequence of well-defined, unambiguous instructions that, when followed, lead to a solution for a given problem or accomplish a particular objective.

45. What is linear and non-linear data structure ?

Ans:- **linear** :- In a linear data structure, elements are arranged in a sequential manner where each element has a unique predecessor and successor, except for the first and last elements.
non-linear :- In a non-linear data structure, elements are not arranged in a sequential manner. There is no strict order or relationship among the elements.

46. What is Primitive and Non- primitive data structure ?

Ans;- Primitive data structures are basic and fundamental data types directly supported by most programming languages. Non-primitive data structures are more complex and are composed of multiple primitive data types. They are not predefined in programming languages but are built using primitive data types..

47. What is Two dimensional array ?

Ans:- A two-dimensional array is a data structure that represents a table or matrix of elements. Unlike a one-dimensional array, which is a linear collection of elements, a two-dimensional array is organized in rows and columns.

48. What is tree traversal algorithm?

Ans:- Tree traversal algorithms are methods used to visit and process all nodes of a tree data structure systematically. There are several ways to traverse a tree, each with its own advantages and use cases.

49. What is the minimum number of nodes that a binary tree can have?

Ans:-A binary tree can have a minimum of zero nodes, which occurs when the nodes have NULL values. Furthermore, a binary tree can also have 1 or 2 nodes.

50. Which sorting algorithm is considered the fastest?

Ans:- There are many types of sorting algorithms: quick sort, bubble sort, balloon sort, radix sort, merge sort, etc. Not one can be considered the fastest because each algorithm is designed for a particular data structure and data set. It would depend on the data set that you would want to sort.

51. What is an AVL tree? (imp)

Ans:- An AVL tree is a type of binary search tree that is always in a state of partially balanced. The balance is measured as a difference between the heights of the subtrees from the root. This self-balancing tree was known to be the first data structure to be designed as such.