Capgemini Technical Interview Questions:

Q1. What do you understand by data structures?

Ans. The data structure is a way that specifies how to organize and manipulate the data. Some examples of Data Structures are arrays, Linked List, Stack, Queue, etc.

Q2. What are some of the applications of Data Structures?

Ans. Compiler Design, Operating System, DBMS, Graphics, Simulation.

Q3. What are the advantages of Linked List over Arrays?

Ans. The size of a linked list can be incremented at runtime which is impossible in the case of the array. The List is dynamically stored in the main memory and grows as per the program demand while the array is statically stored in the main memory, size of which must be declared at compile time.

### Q4. Write the syntax in C to create a node in the singly linked list?

### Ans.

### struct Node

### {

### int data;

### struct Node\* next;

### };

### Q5. What is the use of a doubly-linked list when compared to that of a singly Linked list?

### Ans. Doubly linked list can be traversed in both forward and backward direction. The delete operation in doubly linked list is more efficient if reference to the node to be deleted is given. We can quickly insert a new node before a given node. In singly linked list, to delete a node, reference to the previous node is needed. To get this previous node, sometimes the list is traversed. In Doubly linked list, we can get the previous node using previous reference.

### Q6. What is the difference between an Array and Stack?

### Ans. The main difference between array and stack is accessing of data and traversing of data in Array we can access data by index only or traverse the array from the index where we want but in case of stack we can only access the data in last in first out fashion and traversing is also done from last input to first input only.

### Q7. What are the minimum number of Queues needed to implement the priority queue?

### Ans. Two queues are required to implement the priority queue one for storing the data and other one for priority.

### Q8. What are the different types of traversal techniques in a tree?

### Ans. Three are three types of traversal technique in trees, Preorder Traversal (root -> left -> right) Inorder Traversal (left -> root -> right) Postorder Traversal (left ->right ->root)

### Q9. Why it is said that searching a node in a binary search tree is efficient than that of a simple binary tree?

### Ans. Searching a node in binary search tree is efficient because of property of binary search tree that each left side node is lesser than root node and each right side node is greater than root node so because of this searching will always follow a particular single path there is no need to traverse each node like binary tree.

### Q10. What are the applications of Graph DS?

### Ans. In Computer science graphs are used to represent the flow of computation. Google maps uses graphs for building transportation systems, where intersection of two or more roads are considered to be a vertex and the road connecting two vertices is considered to be an edge, thus their navigation system is based on the algorithm to calculate the shortest path between two vertices. In Facebook, users are considered to be the vertices and if they are friends then there is an edge running between them. Facebook’s Friend suggestion algorithm uses graph theory. Facebook is an example of undirected graph.

### Q11. Can we apply Binary search algorithm to a sorted Linked list?

### Ans. We cannot apply Binary search algorithm to a sorted linked list because binary search works on indices and linked list work on pointer.

### Q12. When can you tell that a Memory Leak will occur?

### Ans. A memory leak occurs when a program loses the ability to free a block of dynamically allocated memory.

### Q13. How will you check if a given Binary Tree is a Binary Search Tree or not?

### Ans. If a node is a left child, then its key and the keys of the nodes in its right subtree are less than its parent’s key. If a node is a right child, then its key and the keys of the nodes in its left subtree are greater than its parent’s key.

### Q14. Which data structure is ideal to perform recursion operation and why?

### Ans. Stack is used to perform recursion operation Because of its LIFO (Last in First Out) property it remembers its 'caller' so knows whom to return when the function has to return. Recursion makes use of system stack for storing the return addresses of the function calls. Every recursive function has its equivalent iterative (non-recursive) function.

### Q15. What are some of the most important applications of a Stack?

### Ans. 1. Infix to Postfix or Infix to Prefix Conversion. 2. Postfix or Prefix Evaluation. 3. Backtracking Procedure. 4. Recursion Procedure.