Name – Sheetal , 7th CSEA2

Roll no.-11172531

Capgemini Technical Interview questions Assignment:

1) What do you mean a data structure?

 A **data structure** is a **data** organization, management, and storage format that enables efficient access and modification. It  is a collection of **data** values, the relationships among them, and the functions or operations that can be applied to the **data**.

2) What are some of the applications of DS?

a) Arrangement of leader-board of a game can be done simply through arrays to store the score and arrange them in descending order to clearly make out the rank of each player in the game.

b) Web pages can be accessed using the previous and the next URL links which are linked using linked list.

c)The music players also use the same technique to switch between music.

d)To keep the track of turns in a multi player game, a [circular linked list](https://www.geeksforgeeks.org/circular-linked-list/) is used.

e) Syntaxes in languages are parsed using stacks.

f) To handle congestion in networking queue can be used.

3)What are the advantages of the linked list over array?

a) linked list has dynamic size whereas for array it is fixed size

b) insertion and deletion is easy in linked list

c)  Less memory consumption as only that many nodes are consumed as many values are to be stored.

d) LinkedList can grow and shrink at run time.

4) Write the syntax in C to create a node in singly linked list.

#include <stdio.h>

#include <stdlib.h>

struct **node**{

int data;

struct **node** \*next;

};

5)What is the use of doubly linked list when compared to a singly linked list?

1) A DLL can be traversed in both forward and backward direction.

2) The delete operation in DLL is more efficient if pointer to the node to be deleted is given.

3) We can quickly insert a new node before a given node

6)What is the difference between an array and stack?

**In an array**, you have a list of elements and you can access any of them at any time. But **in a stack**, there's no random-access operation; there are only Push, Peek and Pop, all of which deal exclusively with the element on the top of the **stack**. A **stack** is data-structure which has a last in first out policy.

7)What are the minimum number of queues need to implement the priority queues?

2 **queues**. one is used for storing data... another is used for **priorities**. **Priority queues** r applied using 2-D array where it has two rows one for element and second for **priority** ,so **minimum numbers of queues** are **needed to implement** are two.

8) What are the different types of traversal techniques in a tree?

**In inorder traversal method**, the left subtree is visited first, then the root and later the right sub-tree. We should always remember that every node may represent a subtree itself.

**In Pre order traversal method,** the root node is visited first, then the left subtree and finally the right subtree.

**In Post order traversal method,** the root node is visited last, hence the name. First we traverse the left subtree, then the right subtree and finally the root node.

9)Why it is said that searching a node in binary search tree is more efficient than the simply binary tree?

**Binary search tree** is data structure which uses the concept of **binary search**. **BST** have a special quality that every **node** has at most 2 **nodes** and left child of every **node** has value **than node** and right child has more value **than node**. By this method it become **easy** to **search** a **node** in a **tree**.

10)What are the applications of graph DS?

* **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**.
* In **World Wide Web**, web pages are considered to be the vertices. There is an edge from a page u to other page v if there is a link of page v on page u. This is an example of **Directed graph**. It was the basic idea .
* In **Operating System**, we come across the Resource Allocation Graph where each process and resources are considered to be vertices. Edges are drawn from resources to the allocated process, or from requesting process to the requested resource. If this leads to any formation of a cycle then a deadlock will occur.

11)Can we apply binary search algorithm to sorted Linked List?

Yes, Binary **search** is possible on the **linked list** if the **list** is **ordered** and **you** know the count of elements in **list**. But While **sorting** the **list**, **you can** access a single element at a time through a pointer to that node i.e. either a previous node or next node.

12)When can you tell that a memory leak will occur?

**Memory leak occurs** when programmers create a memory in heap and forget **to** delete it.

13)How will you check the given binary tree is binary search tree or not?

**To see if a binary tree is a binary search tree:**

* 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.

14)Which data structure is ideal to perform recursion and why?

Many programming languages implement recursion by means of **stacks**. The example of **recursion** as an application of **stack** is keeping books inside the drawer and the removing each book **recursively**.

15)What are some of the most important application of stack?

a) The stack can be used to convert some infix expression into its postfix equivalent, or prefix equivalent.

b)Another great use of stack is during the function call and return process.

c)Stack is used in backtracking process.

16)Convert the below given program to its equivalent prefifix and postfix notations?

Input : Prefix : \*+AB-CD

Output : Postfix : AB+CD-\*

Prefix to Infix : (A+B) \* (C-D)

Infix to Postfix : AB+CD-\*

Input : Postfix : AB+CD-\*

Output : Prefix : \*+AB-CD

Postfix to Infix : (A+B) \* (C-D)

Infix to Prefix : \*+AB-CD

21 Reversing a linked list

**class RevLinkList {**

**static Node head;**

**static class Node {**

**int data;**

**Node next;**

**Node(int d)**

**{**

**data = d;**

**next = null;**

**}**

**}**

**Node reverse(Node node)**

**{**

**Node prev = null;**

**Node current = node;**

**Node next = null;**

**while (current != null) {**

**next = current.next;**

**current.next = prev;**

**prev = current;**

**current = next;**

**}**

**node = prev;**

**return node;**

**}**

**void printList(Node node)**

**{**

**while (node != null) {**

**System.out.print(node.data + " ");**

**node = node.next;**

**}**

**}**

**public static void main(String[] args)**

**{**

**LinkedList list = new LinkedList();**

**list.head = new Node(85);**

**list.head.next = new Node(15);**

**list.head.next.next = new Node(4);**

**list.head.next.next.next = new Node(20);**

**System.out.println("Given Linked list");**

**list.printList(head);**

**head = list.reverse(head);**

**System.out.println("");**

**System.out.println("Reversed linked list ");**

**list.printList(head);**

**}**

**}**

Q24) Find out the Kth smallest element in an unsorted array.

public class KthSamllestElement {

public static void main(String[] args) {

Scanner scr = new Scanner(System.in);

int arr[] = {2,8, 0 ,5 ,6};

Arrays.sort(arr);

System.out.println("Enter k :");

int k = scr.nextInt()-1;

System.out.println("kth smallest number : "+arr[k]);

}

}

Q25) How to find the shortest path between two vertices.

Dijkstra’s algorithm is used to find the shortest path between 2 vertices.

public class Dijkstra {

public static int minDistance(int dist[],boolean inShortestPath[])

{

int min = Integer.MAX\_VALUE;

int min\_index=-1;

for(int i=0;i<dist.length;i++)

{

if(dist[i]<min && !inShortestPath[i])

{

min\_index=i;

min=dist[i];

}

}

return min\_index;

}

public static void print(int dist[],int parent[])

{

System.out.println("Vertex\tDist from source\tParent Vertex");

for(int i=0;i<dist.length;i++)

System.out.println(i+"\t\t"+dist[i]+"\t\t\t"+parent[i]);

}

public static void printShortestPath(int graph[][],int source)

{

int V = graph.length;

int dist[]= new int[V],

parent[]=new int[V];

boolean inShortestPath[] =new boolean[V];

for(int i=0;i<V;i++)

{

dist[i]=Integer.MAX\_VALUE;

parent[i]=-1;

}

parent[0]=0;

dist[0]=0;

for(int i=0;i<V;i++)

{

int u = minDistance(dist,inShortestPath);

inShortestPath[u]=true;

for(int v=0;v<V;v++)

if(graph[u][v]!=0) {

int distance= dist[u]+graph[u][v];

if(!inShortestPath[v] && dist[v]>distance)

{

dist[v]=distance;

parent[v]=u;

}

}

}

print(dist,parent);

}

public static void main(String arg[])

{

int graph[][]= {

{ 0, 4, 0, 0, 0, 0, 0, 8, 0 },

{ 4, 0, 8, 0, 0, 0, 0, 11, 0 },

{ 0, 8, 0, 7, 0, 4, 0, 0, 2 },

{ 0, 0, 7, 0, 9, 14, 0, 0, 0 },

{ 0, 0, 0, 9, 0, 10, 0, 0, 0 },

{ 0, 0, 4, 14, 10, 0, 2, 0, 0 },

{ 0, 0, 0, 0, 0, 2, 0, 1, 6 },

{ 8, 11, 0, 0, 0, 0, 1, 0, 7 },

{ 0, 0, 2, 0, 0, 0, 6, 7, 0 }

};

int source=0;

printShortestPath(graph,source);

}

}