VIVA QUESTIONS

1) Mention what is Linked lists?

A linked list is a data structure that can store a collection of items. In other words, linked lists can be utilized to store several objects of the same type. Each unit or element of the list is referred as a node. Each node has its own data and the address of the next node. It is like a chain. Linked Lists are used to create graph and trees.

2) What type of memory allocation is referred for Linked lists?

Dynamic memory allocation is referred for Linked lists.

3) Mention what is traversal in linked lists?

Term Traversal is used to refer the operation of processing each element in the list.

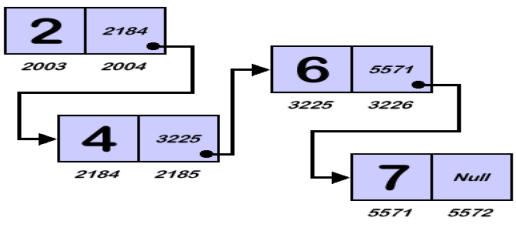
4) Describe what is Node in link list? And name the types of Linked Lists?

Together (data + link) is referred as the Node. Types of Linked Lists are,

- Singly Linked List
- Doubly Linked List
- Circular Linked List
- SLL with header node
- DLL with header node
- Multiple linked list(ex sparse matrix)

5) Mention what is Singly Linked list?

- Singly Linked list are a type of data structure. In a singly linked list, each node in the list stores the contents of the node and a reference or pointer to the next node in the list. It does not store any reference or pointer to the previous node.



6) Mention what is the difference between Linear Array and Linked List?

The difference between Linear Array and Linked List are shown below,

Linear Array	Linked List
Deletion and Insertions are difficult.	Deletion and Insertions can be done easily.
For insertion and deletion, it needs movements	For insertion and deletion, it does not require movement of nodes
In it space is wasted	In it space is not wasted
It is expensive	It is not expensive
It cannot be reduced or extended according to requirements	It can be reduced or extended according to requirements
To avail each element same amount of time is required.	To avail each element different amount of time is required.
In consecutive memory locations elements are stored.	Elements may or may not be stored in consecutive memory locations
We can reach there directly if we have to go to a particular element	To reach a particular node, you need to go through all those nodes that come before that node.

7) Mention what are the applications of Linked Lists?

Applications of Linked Lists are,

1. **Linked lists are used to implement queues, stacks, graphs, etc.**
2. **In Linked Lists you don't need to know the size in advance.**
3. **Linked lists let you insert elements at the beginning and end of the list.**

8) Mention what is the difference between singly and doubly linked lists?

A doubly linked list nodes contain three fields:
An integer value and
Two links to other nodes
one to point to the previous node and
other to point to the next node.
Whereas a singly linked list contains points only to the next node.

9) Define Binary tree

A Tree is a non-linear data structure that helps us hierarchically organize data. The non-linearity that comes with this Data Structure makes it feasible to solve complex problems. A Binary Tree is a special case of tree data structure that can have not more than two children.

10) How can we define the depth or height of a Binary Tree?

The height or depth of a Binary Tree can be defined as the maximum number of nodes in a tree branch. We can also define the maximum number of nodes in a Binary Tree of depth 'x' as '2x - 1'. The depth of the root node is said to be one

11). What are the applications of binary trees?

Some prominent applications of binary trees include
High-bandwidth router for storing router tables.
Algorithms for compression and many more.
Search applications in which data is being left or entered constantly.
Wireless networking.
Memory allocation.

12) Define a Complete binary bree

A complete binary tree is the one in which every level, maybe except for the final, must be filled. Also, all the other nodes must be as far to the left as feasible.

13) Mention three cases that arise when you have found a node that you want to delete from Binary Tree.

Once you have found the node you want to delete, three prominent cases may arise.

1. **The node has no children.**

You need to replace this node with null.

2. **The node has one child.**
   We will replace this node with its only child.
3. **The node has two children.**
   It requires a tree reorganization.

13) Mention differences between the Binary tree and the Binary search tree

Binary Tree	Binary search tree
It is a **non-linear** data structure in which each node has the **utmost two** children, and each of the left and right subtrees should also be a **binary tree**.	It is a **non-linear** data structure in which each node has the **utmost two** children, and each left and right subtree should also be a **binary search tree**.
Elements in this tree are in **random** order.	Elements in this tree are in a **fixed** order. Its left child should have a value **less** than the parent node, and the right child should have a value **greater** than the parent value.
Operations like **searching**, **inserting**, and **deletion** take longer, i.e., **O(n)** time to execute because elements are not sorted.	Operations like **searching**, **inserting**, and **deletion** take shorter, i.e., **O(log n)** time to execute because elements are sorted.
There are **four** types of binary trees. These are **Full** Binary Tree, **Complete** Binary Tree, **Perfect** Binary Tree, and **Extended** Binary Tree.	There are **three** types of binary search trees. These are **AVL** trees, **splay** trees, and **Tango** trees.

14) What is a Threaded Binary Tree?

A tree in which all right child pointers would be null and pointing towards the inorder successor of the node is known as a threaded binary tree.
Similarly, all left child pointers would also be null and pointing towards the inorder predecessor of the node.

15) In what way can you find if the two trees are identical?

Two binary trees are considered identical only when they have the same data and arrangement. For that, we will traverse the tree and compare. You can use the algorithm below for your reference.

Compare data of root node (data1==data2)
Check left subtree recursively.
Call method 'Tree(tree1-> left subtree, tree2-> left subtree)'
Similarly, check the right subtree
if a,b,c are true, return1

16) What is a full binary tree

A full binary tree **is a binary tree where every node has exactly 0 or 2 children.**

Graphs

1. **Define Graph**

A graph is a data structure that represents a collection of interconnected nodes through a set of edges.

This abstract structure is highly versatile and finds applications in various domains, from social network analysis to computer networking.
Core Components

A graph consists of two main components:
Nodes: Also called vertices, these are the fundamental units that hold data.
Edges: These are the connections between nodes, and they can be either directed or undirected.

2.What are different types of graph

Types of Graphs

Undirected**: Edges lack direction, allowing free traversal between connected nodes. Mathematically, (u,v) as an edge implies (v,u) as well.**
Directed (Digraph)**: Edges have a set direction, restricting traversal accordingly. An edge (u,v) doesn't guarantee (v,u).**

Weight Considerations
Weighted**: Each edge has a numerical "weight" or "cost."**
Unweighted**: All edges are equal in weight, typically considered as 1**

Presence of Cycles

1. **Cyclic**: Contains at least one cycle or closed path.
2. **Acyclic**: Lacks cycles entirely.

Edge Density

1. **Dense**: High edge-to-vertex ratio, nearing the maximum possible connections.
2. **Sparse**: Low edge-to-vertex ratio, closer to the minimum.

Connectivity

1. **Connected**: Every vertex is reachable from any other vertex.

2. **Disconnected**: Some vertices are unreachable from others.

Edge Uniqueness

1. **Multigraph**: Allows duplicate edges between vertices.
2. **Simple**: Limits vertices to a single connecting edge.

3) Distinguish between a Tree and a Graph

Graphs and **trees** are both nonlinear data structures, but there are fundamental distinctions between them.

Key Distinctions

- **Uniqueness**: Trees have a single root, while graphs may not have such a concept.
- **Topology**: Trees are **hierarchical**, while graphs can exhibit various structures.
- **Focus**: Graphs center on relationships between individual nodes, whereas trees emphasize the relationship between nodes and a common root.

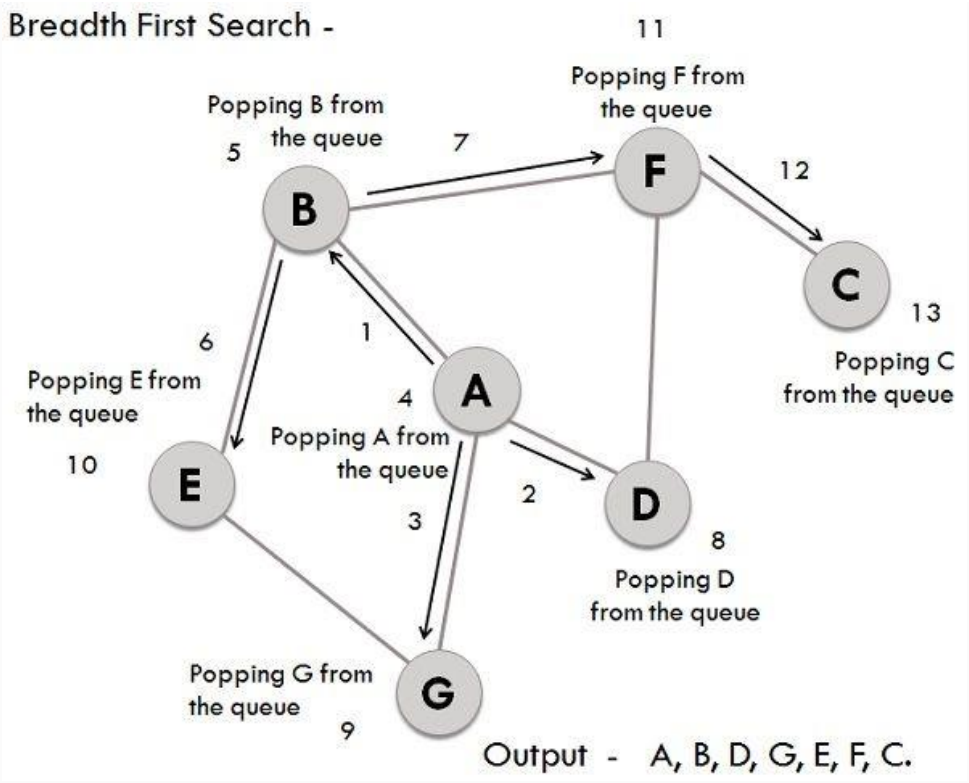
4) Mention two types of graph traversal

DFS

BFS

5) Explain BFS

Breadth-First Search (BFS) is a graph traversal technique that systematically explores a graph level by level. It uses a **queue** to keep track of nodes to visit next and a list to record visited nodes, avoiding redundancy



6) Explain DFS

1. **Depth-First Search** (DFS) is a graph traversal algorithm that's simpler and **often faster** than its breadth-first counterpart (BFS). While it **might not explore all v**

   Initialize: Select a starting vertex, mark it as visited, and put it on a stack.

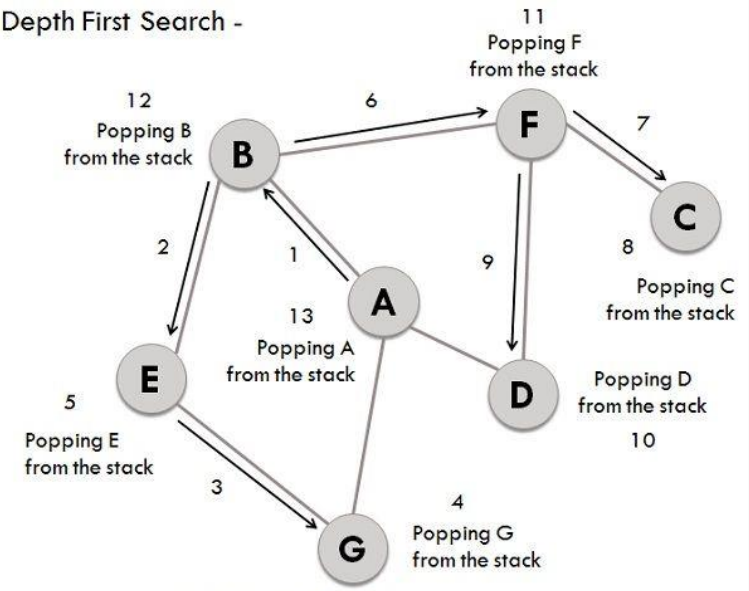
   Loop: Until the stack is empty, do the following:

   - Remove the top vertex from the stack.
   - Explore its unvisited neighbors and add them to the stack.

   Finish: When the stack is empty, the algorithm ends, and all reachable vertices are visited.

vertices, DFS is still fundamental to numerous graph algorithms.

Depth First Search -



Output - A, B, E, G, F, C, D.