**8 Marks Questions**

1. **a) Binomial Heaps:**

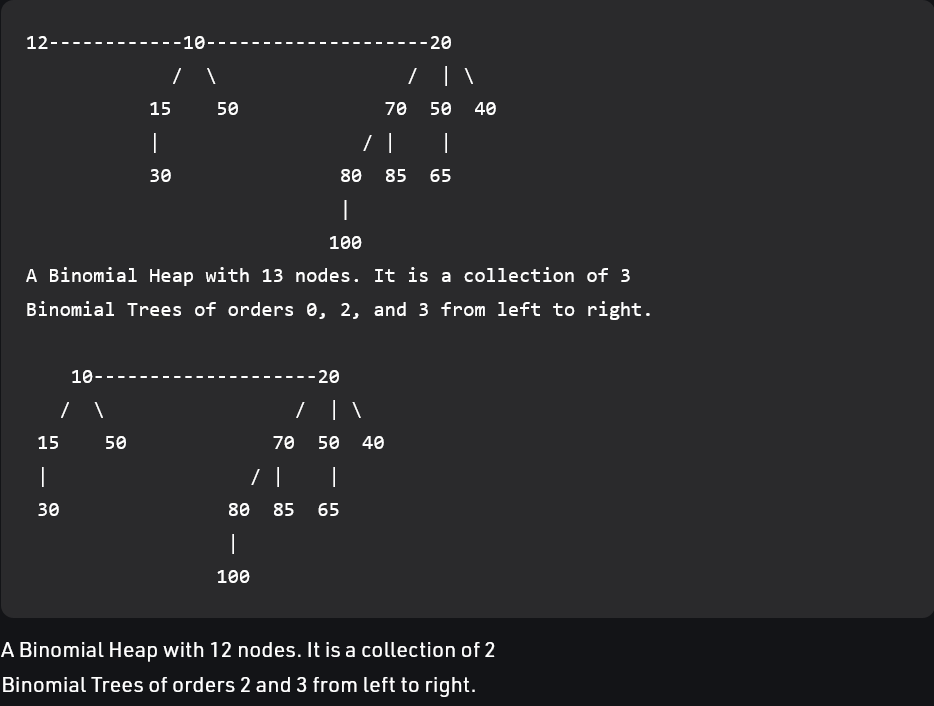
**Ans:** A **Binomial Heap** is a collection of Binomial Trees.

The main application of Binary Heap is as implement a priority queue. Binomial Heap is an extension of Binary Heap that provides faster union or Merge Operation with other operations provided by Binary Heap.

**Operations on Binomial Heap**

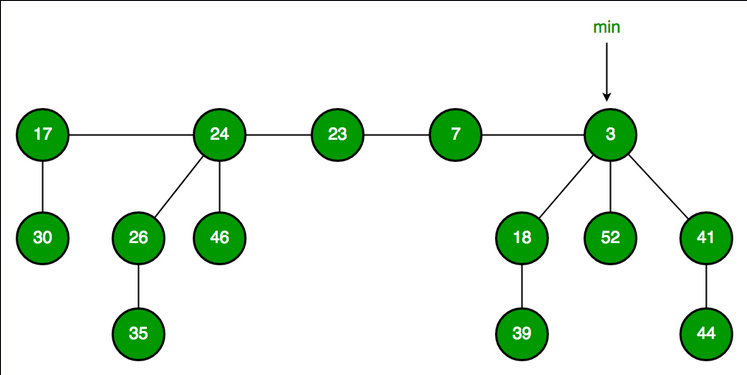
The operations that can be performed on binomial heap are listed as follows -

* Creating a binomial heap
* Finding the minimum key
* Union or merging of two binomial heaps
* Inserting a node
* Extracting minimum key
* Decreasing a key
* Deleting a node



**b) Fibonacci Heaps:**

**Ans:** **Fibonacci Heap** is a collection of trees with min-heap or max-heap property. In Fibonacci Heap, trees can have any shape even all trees can be single nodes (This is unlike Binomial Heap where every tree has to be Binomial Tree).



**Properties of Fibonacci Heap:**

1. It can have multiple trees of equal degrees, and each tree doesn't need to have 2^k nodes.
2. All the trees in the Fibonacci Heap are rooted but not ordered.
3. All the roots and siblings are stored in a separated circular-doubly-linked list.
4. The degree of a node is the number of its children. Node X -> degree = Number of X's children.
5. Each node has a mark-attribute in which it is marked TRUE or FALSE. The FALSE indicates the node has not any of its children. The TRUE represents that the node has lost one child. The newly created node is marked FALSE.
6. The potential function of the Fibonacci heap is F(FH) = t[FH] + 2 \* m[FH].
7. The Fibonacci Heap (FH) has some important technicalities listed below:
   1. min[FH] - Pointer points to the minimum node in the Fibonacci Heap
   2. n[FH] - Determines the number of nodes
   3. t[FH] - Determines the number of rooted trees
   4. m[FH] - Determines the number of marked nodes
   5. F (FH) - Potential Function.
8. **Explain the tree traversal techniques with an example.**

**Ans:**

* **Preorder traversal:**

This technique follows the 'root left right' policy. It means that, first root node is visited after that the left subtree is traversed recursively, and finally, right subtree is recursively traversed. As the root node is traversed before (or pre) the left and right subtree, it is called preorder traversal.

**Algorithm:**

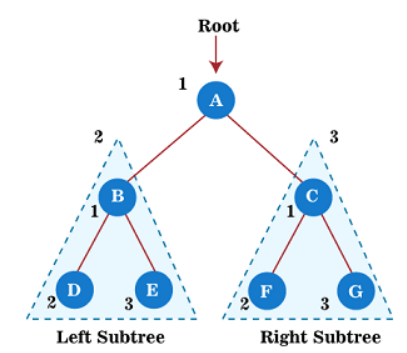
Until all nodes of the tree are not visited

Step 1 - Visit the root node

Step 2 - Traverse the left subtree recursively.

Step 3 - Traverse the right subtree recursively.

**Example:**

* **Postorder traversal:**

This technique follows the 'left-right root' policy. It means that the first left subtree of the root node is traversed, after that recursively traverses the right subtree, and finally, the root node is traversed. As the root node is traversed after (or post) the left and right subtree, it is called postorder traversal.

**Algorithm:**

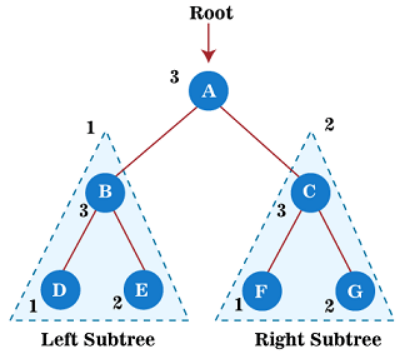
Until all nodes of the tree are not visited

Step 1 - Traverse the left subtree recursively.

Step 2 - Traverse the right subtree recursively.

Step 3 - Visit the root node.

**Example:**

** **

* **Inorder traversal:**

This technique follows the 'left root right' policy. It means that first left subtree is visited after that root node is traversed, and finally, the right subtree is traversed. As the root node is traversed between the left and right subtree, it is named inorder traversal.

**Algorithm:**

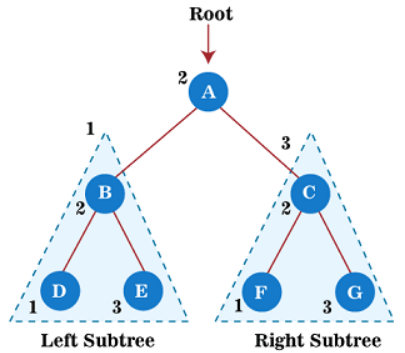
Until all nodes of the tree are not visited

Step 1 - Traverse the left subtree recursively.

Step 2 - Visit the root node.

Step 3 - Traverse the right subtree recursively.

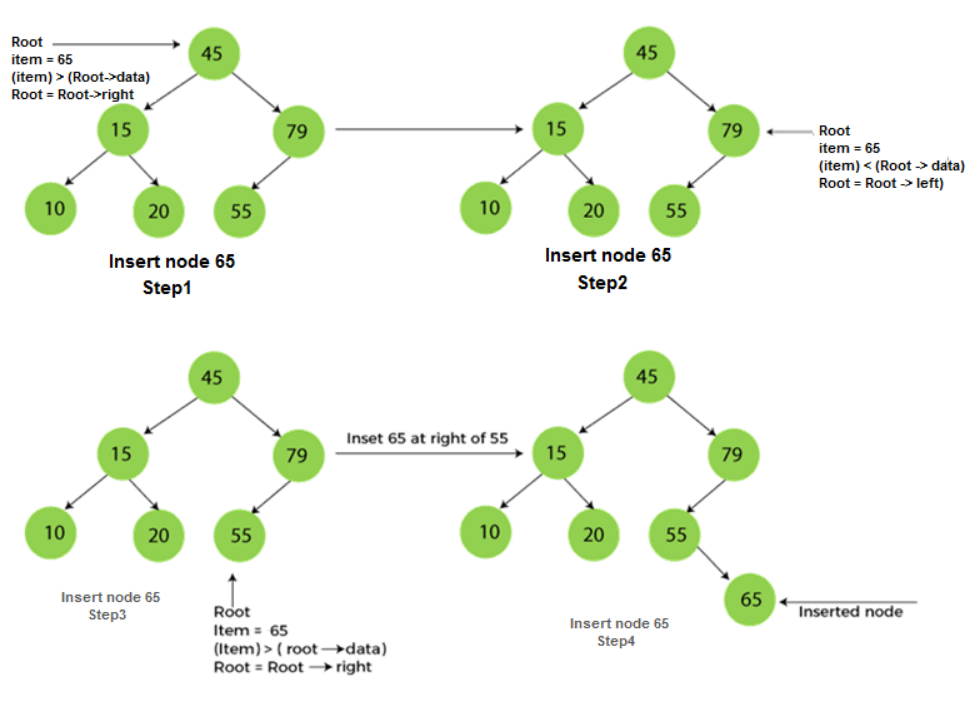
**Example:**

1. **How to insert and delete an element into a binary search tree and write down the code for the insertion routine with an example.**

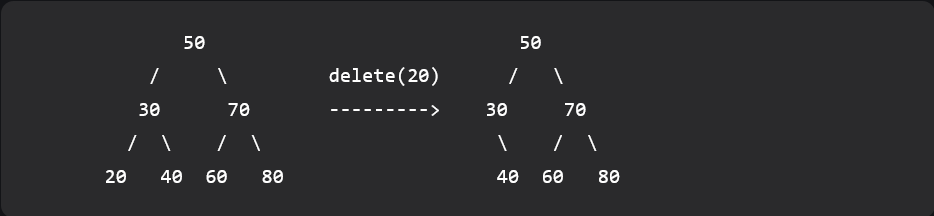
**Ans:**

**Insertion of a key**: A new key in BST is always inserted at the leaf. To insert an element in BST, we have to start searching from the root node; if the node to be inserted is less than the root node, then search for an empty location in the left subtree. Else, search for the empty location in the right subtree and insert the data. Insert in BST is similar to searching, as we always have to maintain the rule that the left subtree is smaller than the root, and right subtree is larger than the root.

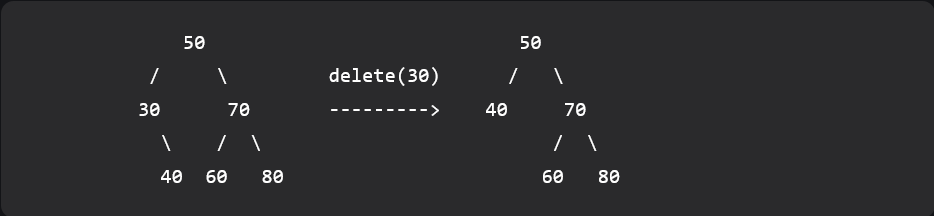


**Deletion in Binary Search tree:** In a binary search tree, we must delete a node from the tree by keeping in mind that the property of BST is not violated. To delete a node from BST, there are three possible situations occur -

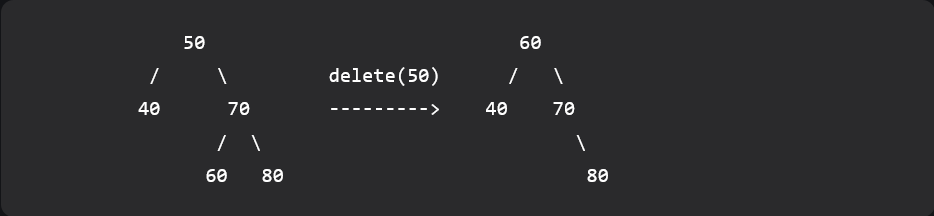
1. The node to be deleted is the leaf node, or,
2. The node to be deleted has only one child, and,
3. The node to be deleted has two children
4. ***Node to be deleted is the*** ***leaf:*** Simply remove from the tree.



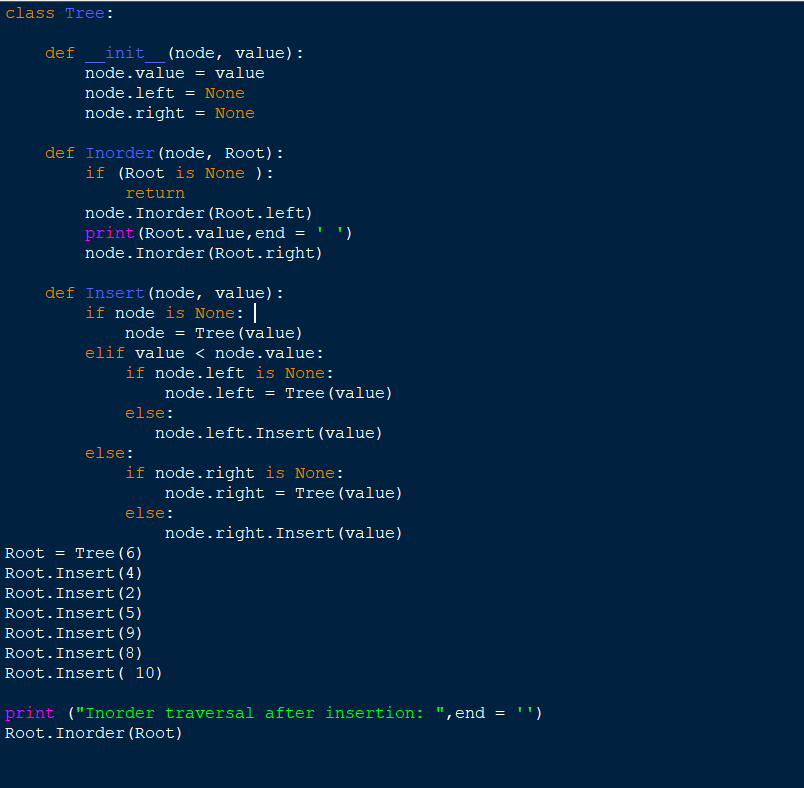
1. ***Node to be deleted has only one child:*** Copy the child to the node and delete the child.



1. ***Node to be deleted has two children:*** Find inorder successor of the node. Copy contents of the inorder successor to the node and delete the inorder successor. Note that inorder predecessor can also be used.



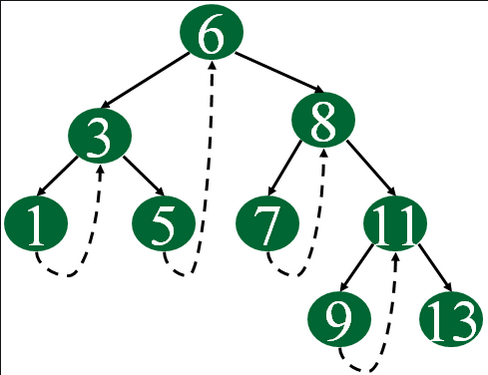
**Code for the insertion routine:**



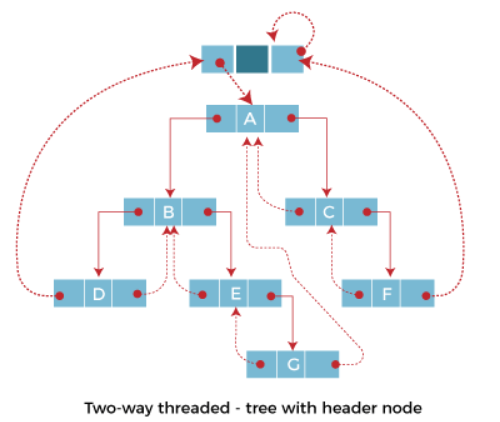
1. **What are threaded binary tree? Write an algorithm for inserting a node in a threaded binary tree.**

**Ans:** The idea of threaded binary trees is to make inorder traversal faster and do it without stack and without recursion. A binary tree is made threaded by making all right child pointers that would normally be NULL point to the inorder successor of the node (if it exists).

***Single Threaded:*** Where a NULL right pointers is made to point to the inorder successor (if successor exists).



***Double Threaded:*** Where both left and right NULL pointers are made to point to inorder predecessor and inorder successor respectively. The predecessor threads are useful for reverse inorder traversal and postorder traversal.



**Algorithm for inserting a node in a threaded binary tree:**

**Case 1: Insertion in empty tree**

Both left and right pointers of **tmp** will be set to NULL and new node becomes the root.



**Case 2: When new node inserted as the left child**

After inserting the node at its proper place we have to make its left and right threads points to inorder predecessor and successor respectively. Before insertion, the left pointer of parent was a thread, but after insertion it will be a link pointing to the new node.

**Case 3: When new node is inserted as the right child**

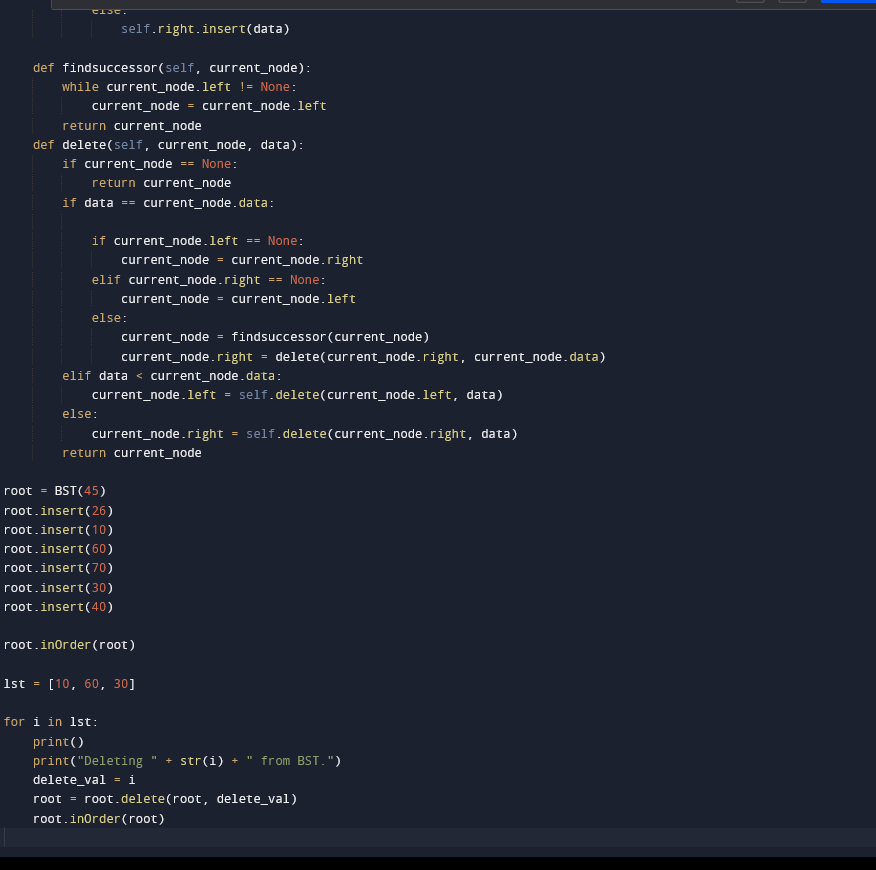
The parent of tmp is its inorder predecessor. The node which was inorder successor of the parent is now the inorder successor of this node tmp. Before insertion, the right pointer of parent was a thread, but after insertion it will be a link pointing to the new node.

Explanation: {<https://www.geeksforgeeks.org/threaded-binary-tree-insertion/>}

1. **Create a binary search tree for the following numbers start from an empty binary search tree. 45,26,10,60,70,30,40 Delete keys 10, 60 and 45 one after the other and show the trees at each stage.**

**Ans:**

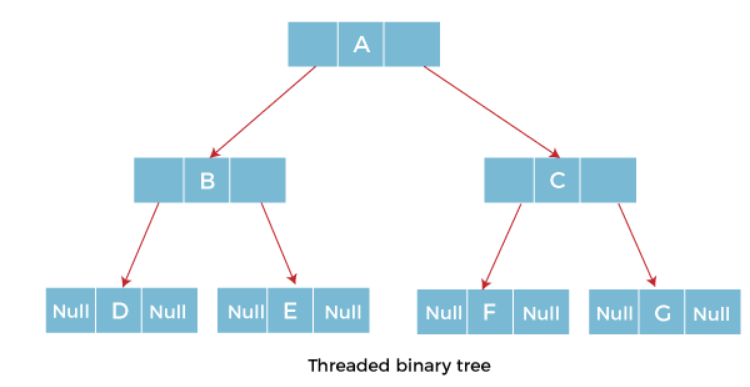


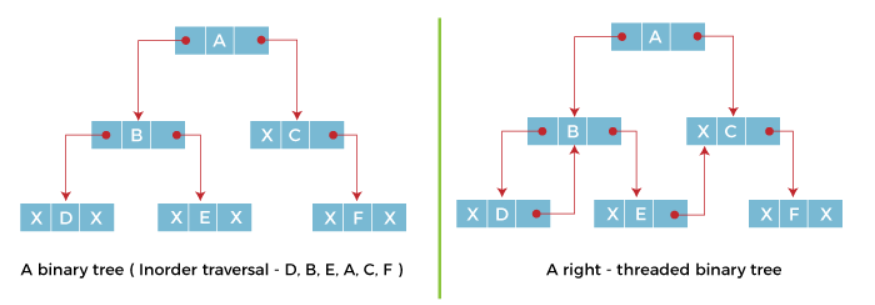


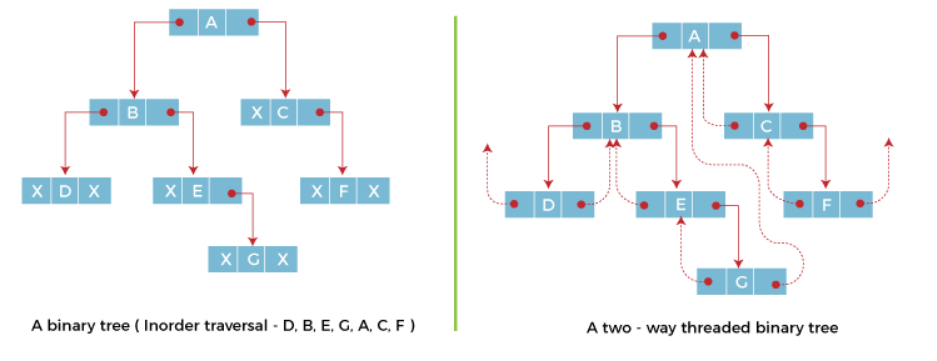
**5 Marks Questions**

* 1. **Explain the logic behind using “Threaded binary tree” in Data Structures. Draw a labeled diagram to show working of threaded binary tree.**

**Ans:** In the linked representation of binary trees, more than one half of the link fields contain NULL values which results in wastage of storage space. If a binary tree consists of **n** nodes then **n+1** link fields contain NULL values. So in order to effectively manage the space, a method was devised by Perlis and Thornton in which the NULL links are replaced with special links known as threads. Such binary trees with threads are known as **threaded binary trees**. Each node in a threaded binary tree either contains a link to its child node or thread to other nodes in the tree.







1. **Describe Binary Tree along with its representation. How will you search an element in Binary Tree? Explain.**

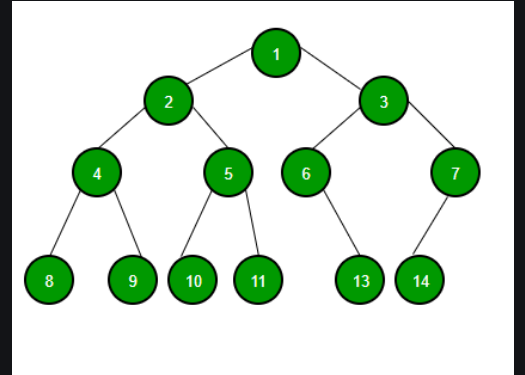
**Ans:**

*“””****Binary Tree*** *is defined as a Tree data structure with at most 2 children. Since each element in a binary tree can have only 2 children, we typically name them the left and right child.”””*

**Binary Tree Representation**  
A Binary tree is represented by a pointer to the topmost node of the tree. If the tree is empty, then the value of the root is NULL.

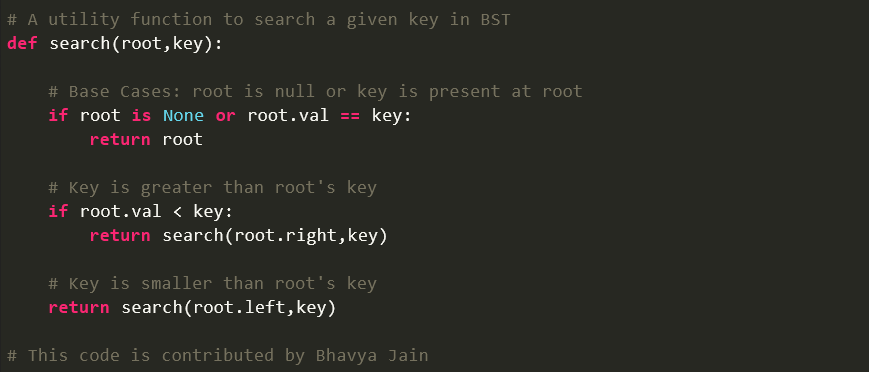
Binary Tree node contains the following parts:

1. Data
2. Pointer to left child
3. Pointer to right child



**Search an element in Binary Tree:**

* Start from the root.
* Compare the searching element with root, if less than root, then recursively call left subtree, else recursively call right subtree.
* If the element to search is found anywhere, return true, else return false.



1. **Explain the following:-**

**Ans:**

**Binary Tree:** *“””Binary Tree is defined as a Tree data structure with at most 2 children. Since each element in a binary tree can have only 2 children, we typically name them the left and right child.”””*

\*\*\* Same as Answer 2 (Page - 11) \*\*\*

**Binary Search Tree:**

A binary Search Tree is a node-based binary tree data structure which has the following properties:

* The left subtree of a node contains only nodes with keys lesser than the node’s key.
* The right subtree of a node contains only nodes with keys greater than the node’s key.
* The left and right subtree each must also be a binary search tree.   
  There must be no duplicate nodes.

**Complete Binary Tree:**

A complete binary tree is a special type of binary tree where all the levels of the tree are filled completely except the lowest level nodes which are filled from as left as possible.

\*\* Same Thing In Depth \*\*

A Complete Binary Tree is a type of binary tree that satisfies some particular set of conditions. These conditions are:

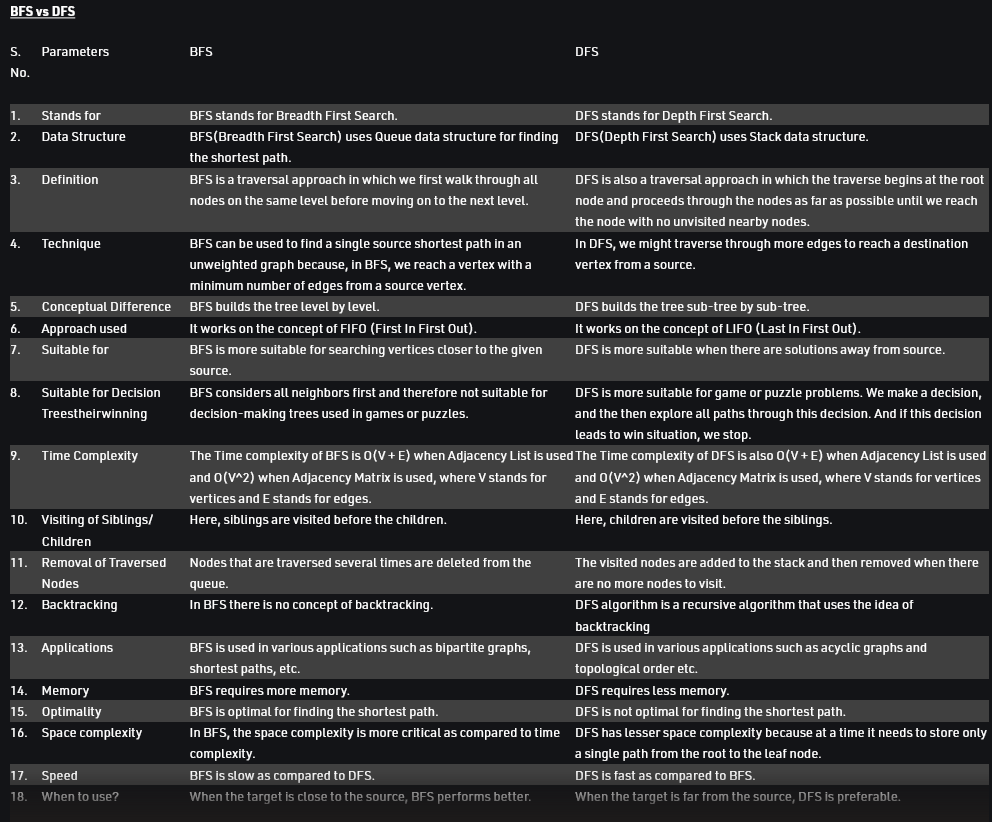
* Each level in the complete binary should be completely filled only other than the last level of the Complete Binary Tree. Complete filling of a particular level means that each parent that is present in that particular level should have exactly two children nodes, i.e., the left node and the right node. If any of the parents in a level doesn't have both the right child and the left then it will not be considered as completely filled.
* Another condition that is required for a binary to satisfy in order to be called a Complete Binary Tree is that the last level of the tree should have all the keys as left as possible. That means if a parent node is present in the last level of a Complete Binary Tree then it should have the child as only the left child node.

1. **Write Kruskal’s Algorithm for finding Minimum Spanning Tree.**

**Ans: Explanation:** {<https://www.geeksforgeeks.org/kruskals-minimum-spanning-tree-algorithm-greedy-algo-2/>}

1. **Outline the distinguishing features of Depth First Search (DFS) and Breadth First Search (BFS).**

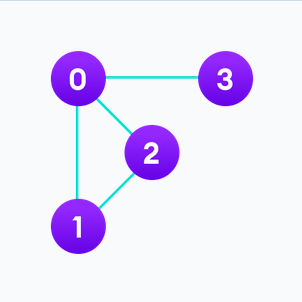
**Ans:**



1. **Explain Adjacency Matrix with the help of suitable diagram.**

**Ans:** An adjacency matrix is a way of representing a graph as a matrix of booleans (0's and 1's). A finite graph can be represented in the form of a square matrix on a computer, where the Boolean value of the matrix indicates if there is a direct path between two vertices.

**Example:**



Each cell in the above table/matrix is represented as Aij, where i and j are vertices. The value of Aij is either 1 or 0 depending on whether there is an edge from vertex i to vertex j.

If there is a path from i to j, then the value of Aij is 1 otherwise its 0. For instance, there is a path from vertex 1 to vertex 2, so A12 is 1 and there is no path from vertex 1 to 3, so A13 is 0.

In case of undirected graphs, the matrix is symmetric about the diagonal because of every edge (i,j), there is also an edge (j,i).

1. **How will you detect a cycle in a Directed as well as Undirected graph? Explain with help of an example.**

**Ans:**

<https://youtu.be/0dJmTuMrUZM>

<https://youtu.be/L0DcePeWHnM>

1. **What are various representations of Binary Tree?**

**Ans:** Let T be a Binary Tree. There are two ways of representing T in the memory as follow

* 1. Sequential Representation of Binary Tree.
  2. Link Representation of Binary Tree.

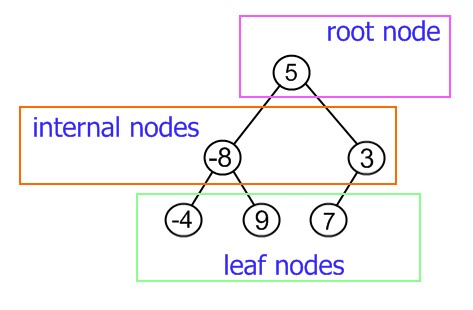
1. **Write Advantages of Threaded Binary Tree.**

**Ans:** If a binary tree consists of **n** nodes then **n+1** link fields contain NULL values. So in order to effectively manage the space, a method was devised by Perlis and Thornton in which the NULL links are replaced with special links known as threads. Such binary trees with threads are known as **threaded binary trees**.

A threaded tree is more efficient because it can traverse the tree more quickly. A binary tree can take a long time to traverse the tree because it needs to search through every node.

1. **Define Leaf.**

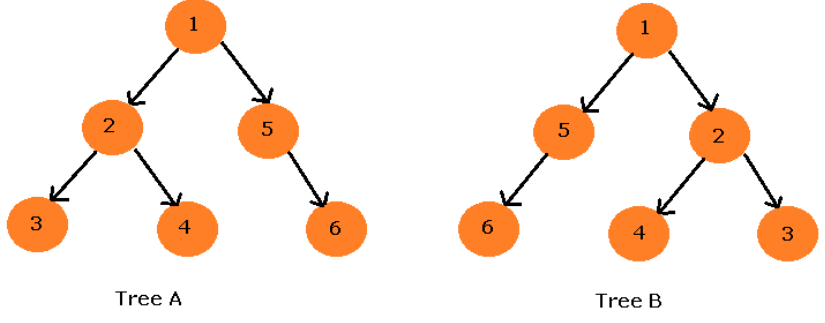
**Ans:**



Very last nodes of every sub-tree are leaf nodes. Write in your own language.

1. **What is ordered Tree?**

**Ans:** The name suggests that the ordered tree must be an organized tree in which nodes are in some order.



In the figure above, for any node, its left child has less value than the node itself and the right child has a value greater than the node.

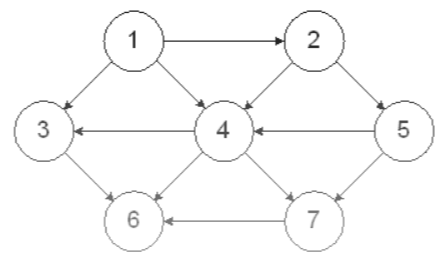
1. **Define Graph. List any 3 application area of Graph.**

**Ans:** A graph is a non-linear data structure, which consists of vertices (or nodes) connected by edges (or arcs) where edges may be directed or undirected.

* 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**.

1. **Define Out Degree of Graph.**

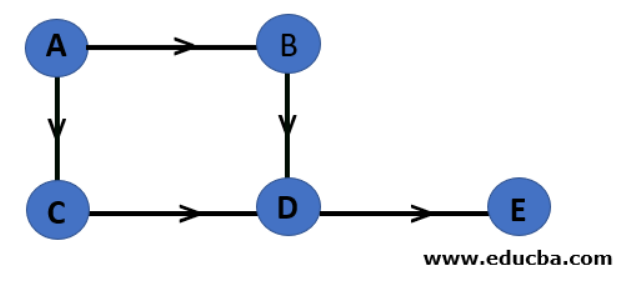
**Ans:**



Out degree is number of arrows leaving from a node. (The arrow heads points away from the node).

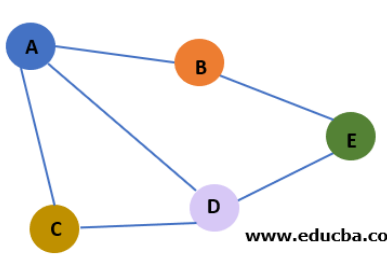
1. **Define Undirected Graph.**

**Ans:** A directed graph is one where the edges can be traversed in a specified direction only.



1. **Define Adjacent Nodes.**

**Ans:**



Two nodes are called adjacent if they are connected through an edge. Node A is adjacent to nodes B, C, and D in the above example, but not to node E.