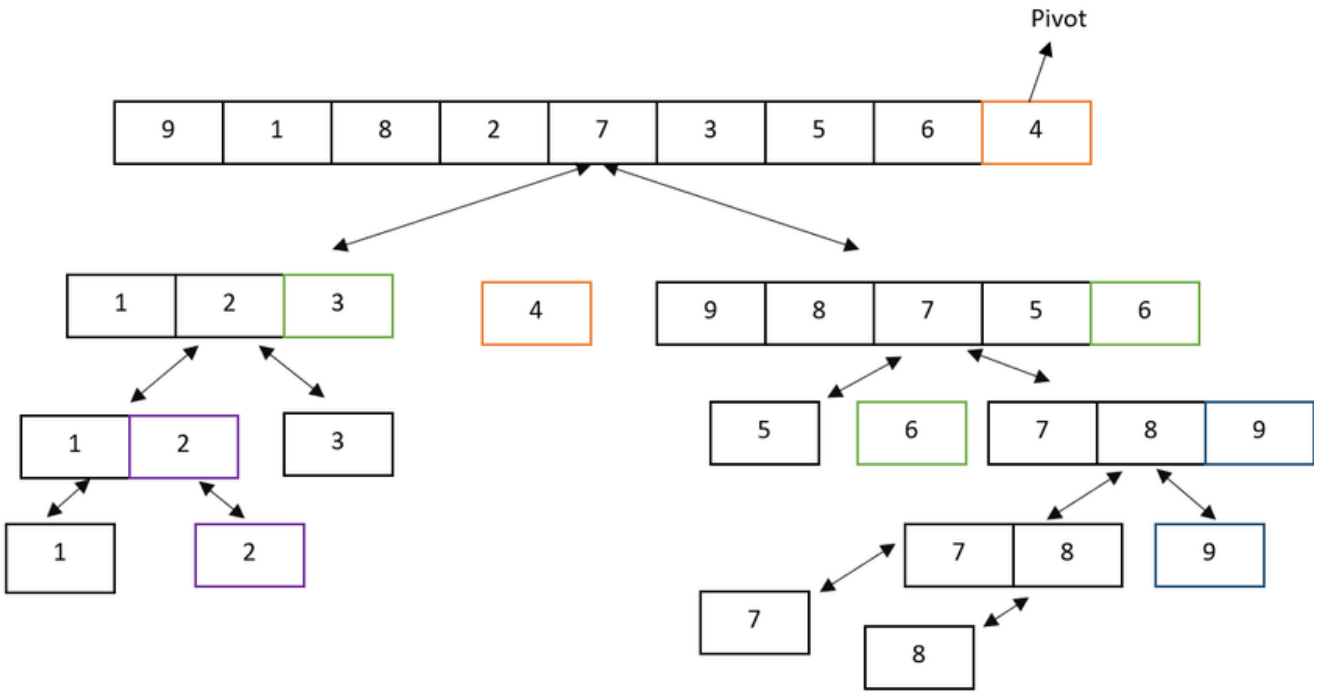
Data structures an Algorithms

Quicksort

- Quicksort is one of the most efficient sorting algorithms. It works by breaking an array (partition) into smaller ones and swapping (exchanging) the smaller ones, depending on a comparison with the 'pivot' element picked.



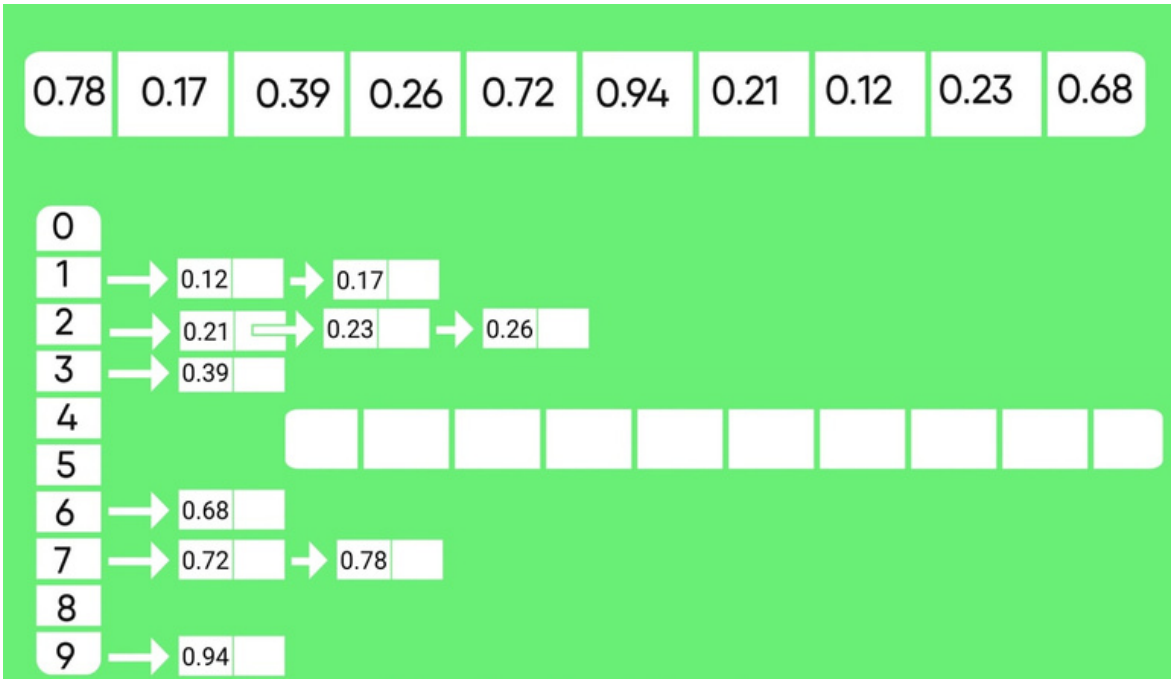
Complexity Analysis of Quicksort:

- Quicksort's best-case time complexity is O (n*logn)
- It has a worst-case time complexity of O(n^2), which occurs when the pivot is chosen poorly.

Data structures an Algorithms

Bucket Sort

- Bucket sort is a sorting algorithm that separate the elements into multiple groups said to be buckets. Elements in bucket sort are first uniformly divided into groups called buckets, and then they are sorted by any other sorting algorithm. After that, elements are gathered in a sorted manner.



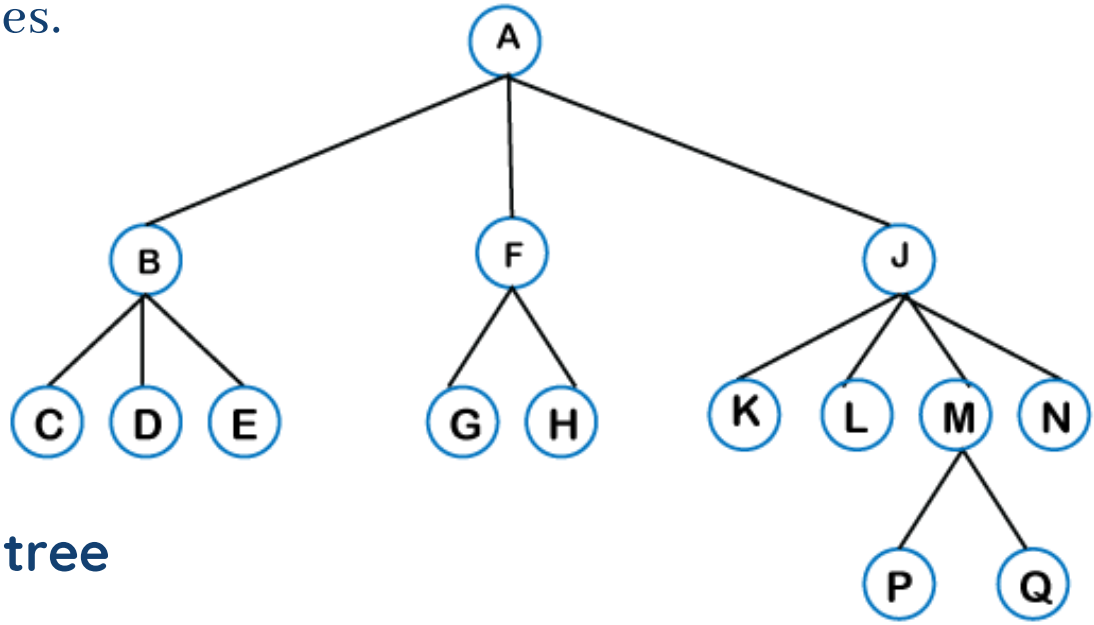
Complexity Analysis of Quicksort:

worst casr :- O(n2)
best case :- O(n + k)

Data structures an Algorithms

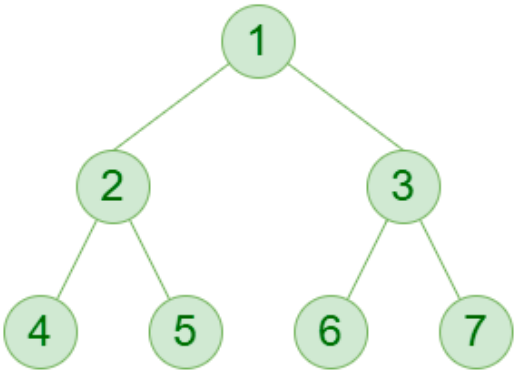
General tree:

In the general tree, a node can have either 0 or maximum n number of nodes. There is no restriction imposed on the degree of the node (the number of nodes that a node can contain). The topmost node in a general tree is known as a root node. The children of the parent node are known as subtrees.



Binary tree

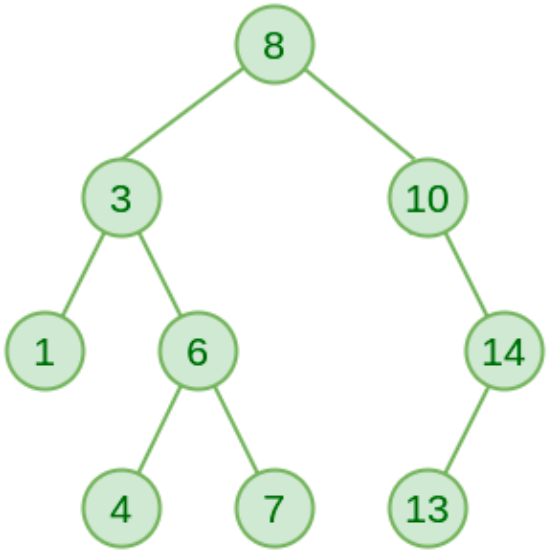
The binary tree is the kind of tree in which most two children can be found for each parent. The kids are known as the left kid and right kid. This is more popular than most other trees.



Data structures an Algorithms

Binary Search Tree (BST) :

Binary Search Tree (BST) is a binary tree extension with several optional restrictions. The left child value of a node should in BST be less than or equal to the parent value, and the right child value should always be greater than or equal to the parent's value.This Binary Search Tree property makes it ideal for search operations since we can accurately determine at each node whether the value is in the left or right sub-tree.



Advantages of Tree data structure

- Efficient insertion, deletion, and search operations.
- Trees are flexibility in terms of the types of data that can be stored.
- It is used to represent hierarchical relationships.
- It has the ability to represent a recursive structure.
- Trees are dynamic in nature.
- Tree data structures can automatically self-organize as new data is added or removed, which can improve performance and reduce complexity.

Disadvantages of Tree data structure

- Trees require additional memory for pointers.
- Trees are not the best choice for data that does not have hierarchical relationships.

Data structures an Algorithms

Basic Terminologies In Tree Data Structure:

- **Root:**
  - The root node is the topmost node in the tree hierarchy. In other words, the root node is the one that doesn't have any parent. In the above structure, node numbered 1 is the root node of the tree. If a node is directly linked to some other node, it would be called a parent-child relationship.

- **Child node:**
  - If the node is a descendant of any node, then the node is known as a child node.

- **Parent :**
  - If the node contains any sub-node, then that node is said to be the parent of that sub-node.

- **Sibling :**
  - The nodes that have the same parent are known as siblings.

- **Leaf Node:**
  - The node of the tree, which doesn't have any child node, is called a leaf node. A leaf node is the bottom-most node of the tree. There can be any number of leaf nodes present in a general tree. Leaf nodes can also be called external nodes.

- **Internal nodes:**
  - A node has atleast one child node known as an internal

- **Ancestor node:**
  - An ancestor of a node is any predecessor node on a path from the root to that node. The root node doesn't have any ancestors. In the tree shown in the above image, nodes 1, 2, and 5 are the ancestors of node 10.

Data structures an Algorithms

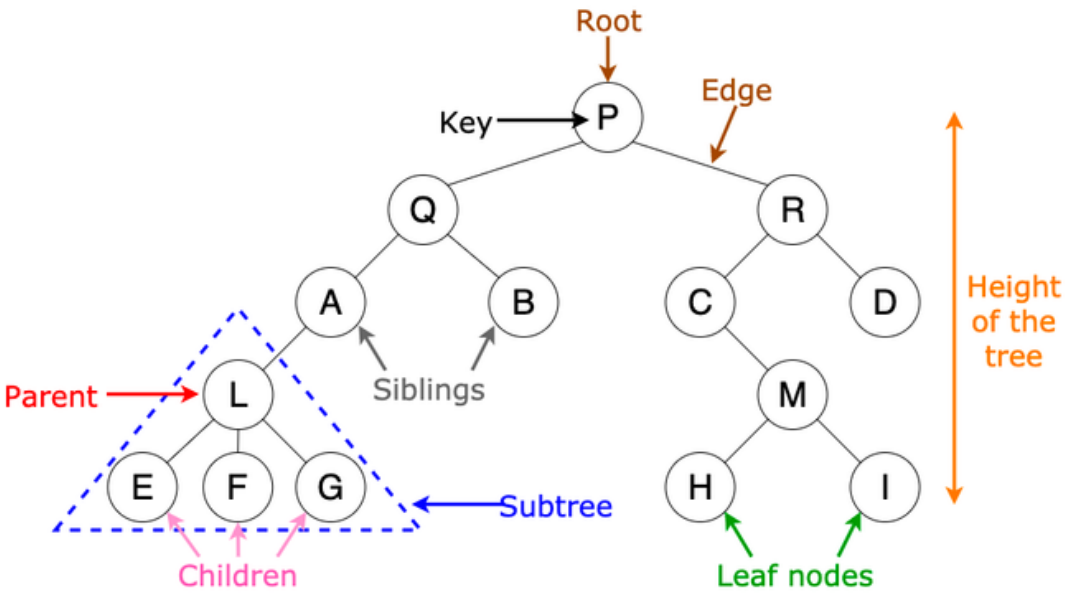
Basic Terminologies In Tree Data Structure:

- **Number of edges:**
  - An edge can be defined as the connection between two nodes. If a tree has N nodes then it will have (N-1) edges. There is only one path from each node to any other node of the tree.

- **Height of a node:**
  - The height of a node can be defined as the length of the longest path from the node to a leaf node of the tree.

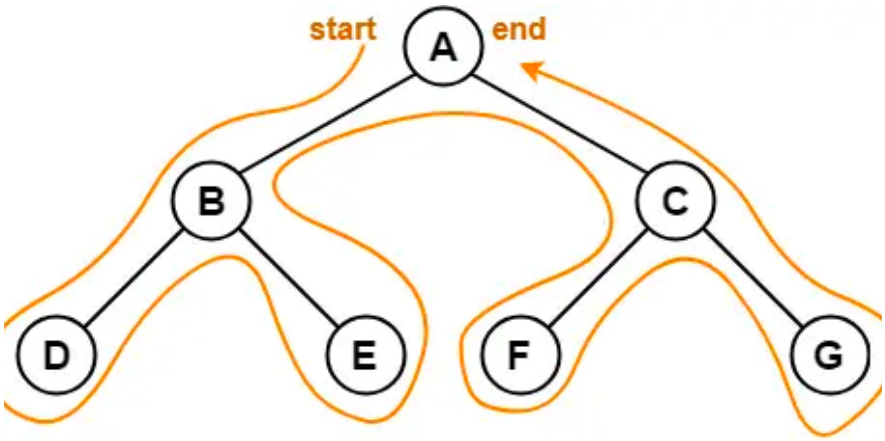
- **Height of the Tree:**
  - The height of a tree is the length of the longest path from the root of the tree to a leaf node of the tree.

- **Degree of a Node:**
  - The total count of subtrees attached to that node is called the degree of the node. The degree of a leaf node must be 0. The degree of a tree is the maximum degree of a node among all the nodes in the tree.



Data structures an Algorithms

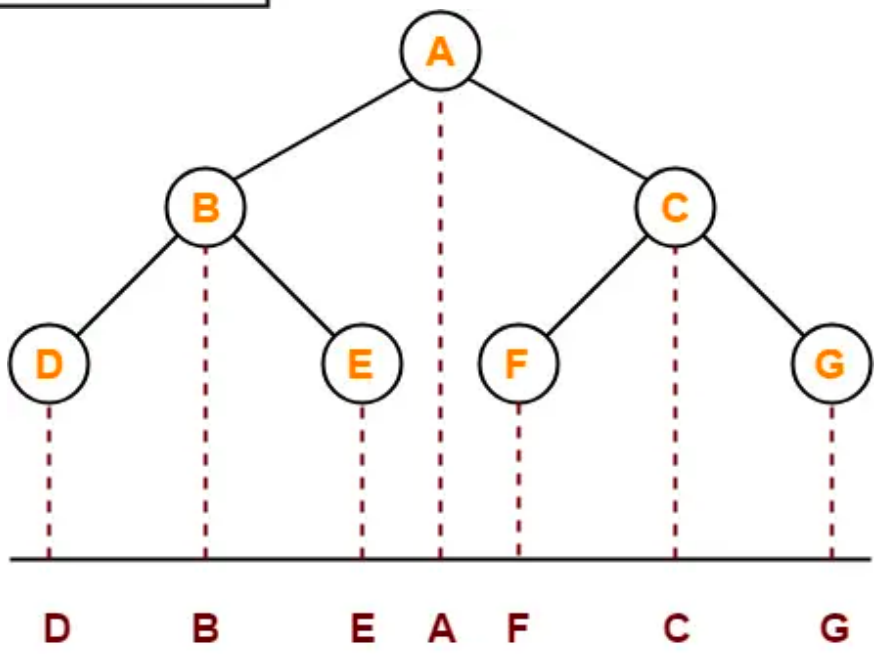
Tree Traversal



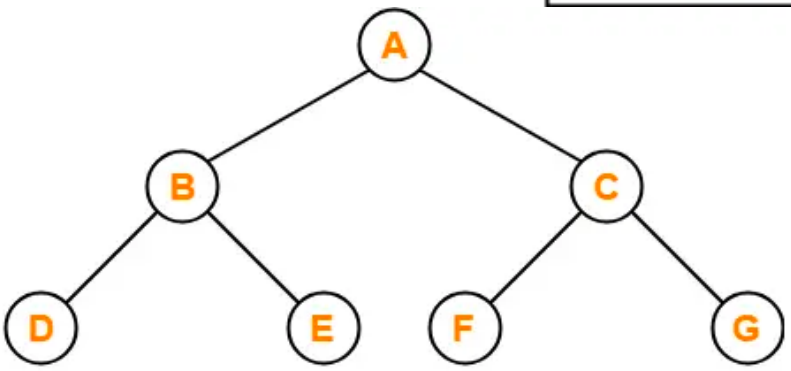
Root → Left → Right

Preorder Traversal : A , B , D , E , C , F , G

Left → Root → Right



Inorder Traversal : D , B , E , A , F , C , G



Left → Right → Root

Postorder Traversal : D , E , B , F , G , C , A

KEEP LEARNING AND HAPPY CODING