



# DATA STRUCTURES

[HOME](#)[SUBJECTS](#)[DOWNLOADS](#)[AUTHORS](#)[CONTACT US](#)

## UNIT 1

Introduction to  
Algorithm

Performance Analysis

Space Complexity

Time Complexity

Asymptotic Notations

Linear & Non-Linear

Data Structures

Single Linked List

Circular Linked List

Double Linked List

Arrays

Sparse Matrix

## UNIT 2

Stack ADT

Stack Using Array

Stack Using Linked  
List

Expressions

Infix to Postfix

Postfix Evaluation

Queue ADT

Queue Using Array



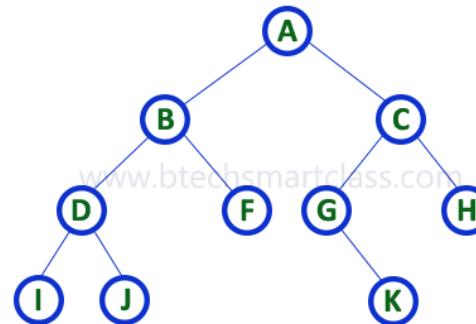
## Binary Tree Representations



A binary tree data structure is represented using two methods. Those methods are as follows...

1. Array Representation
2. Linked List Representation

Consider the following binary tree...



### 1. Array Representation

In array representation of binary tree, we use a one dimensional array (1-D Array) to represent a binary tree.

Consider the above example of binary tree and it is represented as follows...

Queue Using Linked List

Circular Queue

Double Ended Queue

### UNIT 3

Tree - Terminology

Tree Representations

Binary Tree

Binary Tree

Representations

Binary Tree

Traversals

Threaded Binary trees

Max Priority Queue

Max Heap

Introduction to

Graphs

Graph

Representations

Graph Traversal - DFS

Graph Traversal - BFS

### UNIT 4

Linear Search

Binary Search

Hashing

Insertion Sort

Selection Sort

Radix Sort

Quick Sort

Heap Sort

Comparison of

Sorting Methods



To represent a binary tree of depth 'n' using array representation, we need one dimensional array with a maximum size of  $2^{n+1} - 1$ .

## 2. Linked List Representation

We use double linked list to represent a binary tree. In a double linked list, every node consists of three fields. First field for storing left child address, second for storing actual data and third for storing right child address.

In this linked list representation, a node has the following structure...



The above example of binary tree represented using Linked list representation is shown as follows...

**UNIT 5**

Binary Search Tree

AVL Trees

B - Trees

Red - Black Trees

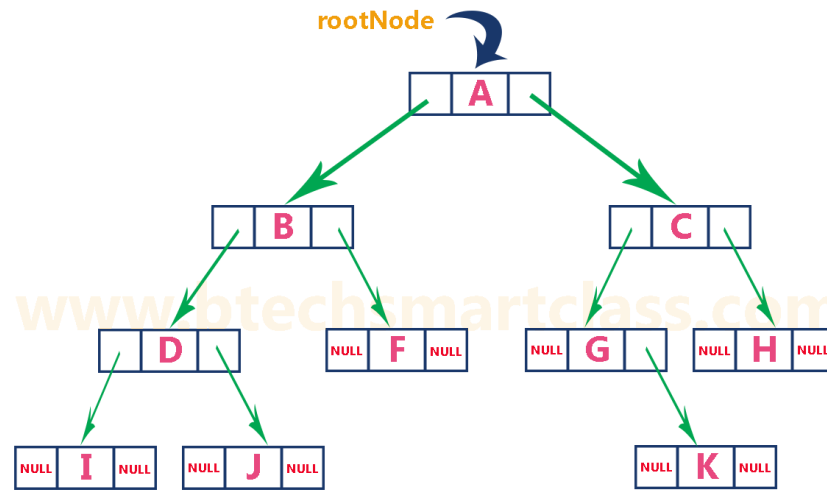
Splay Trees

Comparison of Search  
Trees

Knuth-Morris-Pratt

Algorithm

Tries

[About Us](#) | [Contact Us](#)

Website designed by RAJINIKANTH B