

Binary Tree

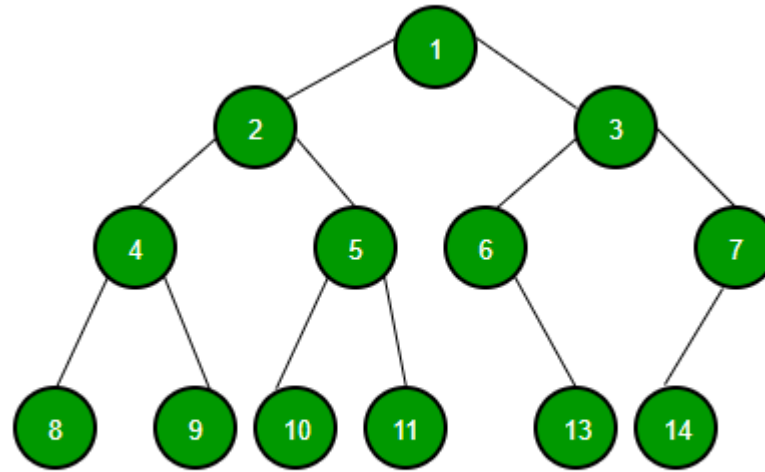
A binary tree is a tree data structure composed of nodes, each of which has at most, two children, referred to as left and right nodes and the tree begins from root node.

Representation of Binary Tree:

Each node in the tree contains the following:

- Data
- Pointer to the left child
- Pointer to the right child





This is the most typical representation of Binary Tree, but we have some other representation of trees as well, For example Binary Heap and Segment tree can be easily represented using Arrays. But in general when we talk about Binary tree or Binary Search Tree, they are typically represented using the linked structure.

Why do we study Binary Tree and not general n-array tree, where a node can have n number off children?

The reason is the most popular tree data structure that is used a lot is based on Binary Tree. Binary Search Tree is the most used tree data structure and it is a binary tree variation. Binary Heap is also a popular tree data structure, which is used to represent priority queues, is also a binary tree data structure. Segment tree data structure which is used for range queries is also a binary tree data structure. So most of the practically used and popular data structure are Binary Tree variation, that is why we study binary tree most if the times in academics.

In C++/Java, we can use classes as part of their OOP feature. Below is an example of a tree node with integer data.

C++

Java

```
// Class containing left and right child
// of current node and key value
class Node {
    int key;
    Node left, right;

    public Node(int item)
    {
        key = item;
        left = right = null;
    }
}
```

Implementation of Binary Tree:

Let us create a simple tree with 4 nodes. The created tree would be as follows.

```
    1    <-- root
   /  \
  2    3
 /
4
```

Below is the Implementation of the binary tree:

C++

Java



Dash



All



Articles



Videos



Problems



Quiz



```
// Class containing left and right child
// of current node and key value

class Node {

    int key;

    Node left, right;

    public Node(int item)

    {

        key = item;

        left = right = null;

    }

}

// A Java program to introduce Binary Tree

class BinaryTree {

    // Root of Binary Tree

    Node root;
```





Dash



All



Articles



Videos



Problems



Quiz

Get 90% Refund!

Courses

Tutorials

Jobs

Practice

Contests



<<

>>

// Constructors

BinaryTree(int key) { root = new Node(key); }

BinaryTree() { root = null; }

public static void main(String[] args)

{

BinaryTree tree = new BinaryTree();

// create root

tree.root = new Node(1);

/* following is the tree after above statement

null null */

tree.root.left = new Node(2);

tree.root.right = new Node(3);

/* 2 and 3 become left and right children of 1



P



Dash



All



Articles



Videos



Problems



Quiz



```

      1
     /  \
    2    3
   / \  / \
null null null null */

```

```
tree.root.left.left = new Node(4);
```

```
/* 4 becomes left child of 2
```

```

      1
     /  \
    2    3
   / \  / \
  4  NULL NULL NULL
 / \
NULL NULL

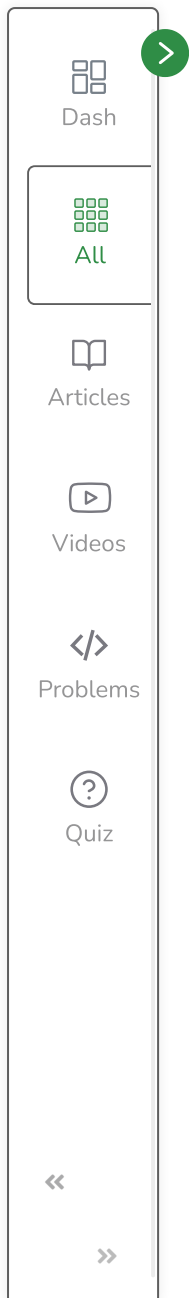
```

```

*/
}
}

```



[Mark as Read](#)[🚩 Report An Issue](#)

If you are facing any issue on this page. Please let us know.

