

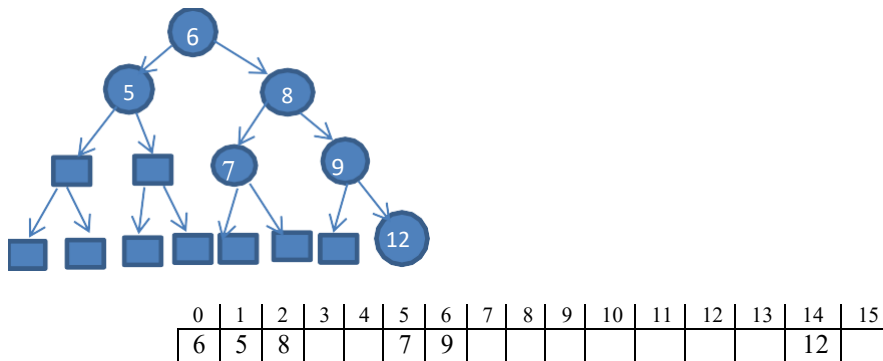
## Binary Search Trees (BST)

### Objective

The objective of this lab is to understand Binary Search Tree using array based implementation. Examine how nonlinear data structure can be stored in linear memory.

### Task

1. Create a binary search tree as shown below and perform insert, display and find operations. Use constructor as needed.



### Algorithm to Create binary tree

1. Compare VALUE with root node N of tree
  - If VALUE < N, proceed to the left child of N
  - If VALUE ≥ N, proceed to the right child of N
2. Repeat step 1. until N=null and assign it a new node with value VALUE.

### Procedure

- Array implementation of the BST that perform insert, find and display operations. Where formula for left-child is  $2P+1$  and formula for right child is  $2P+2$ .

```

Class BST<T extends Comparable<T>>{
    T[] tree;
    BST(int size){
        Tree=(T[]) new Comparable[size];
    }

    public void insert(T d){ code here }

    public void find(T d){ code here }

    public void delete(T d) { code here }

    public void traverse(){ code here }
}
    
```