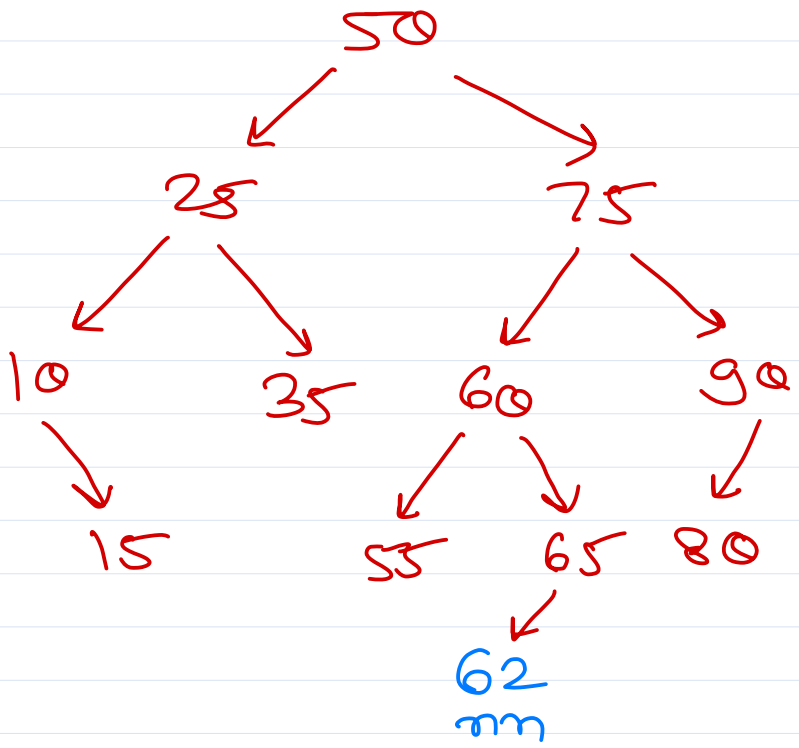
Data Structure & Algorithms

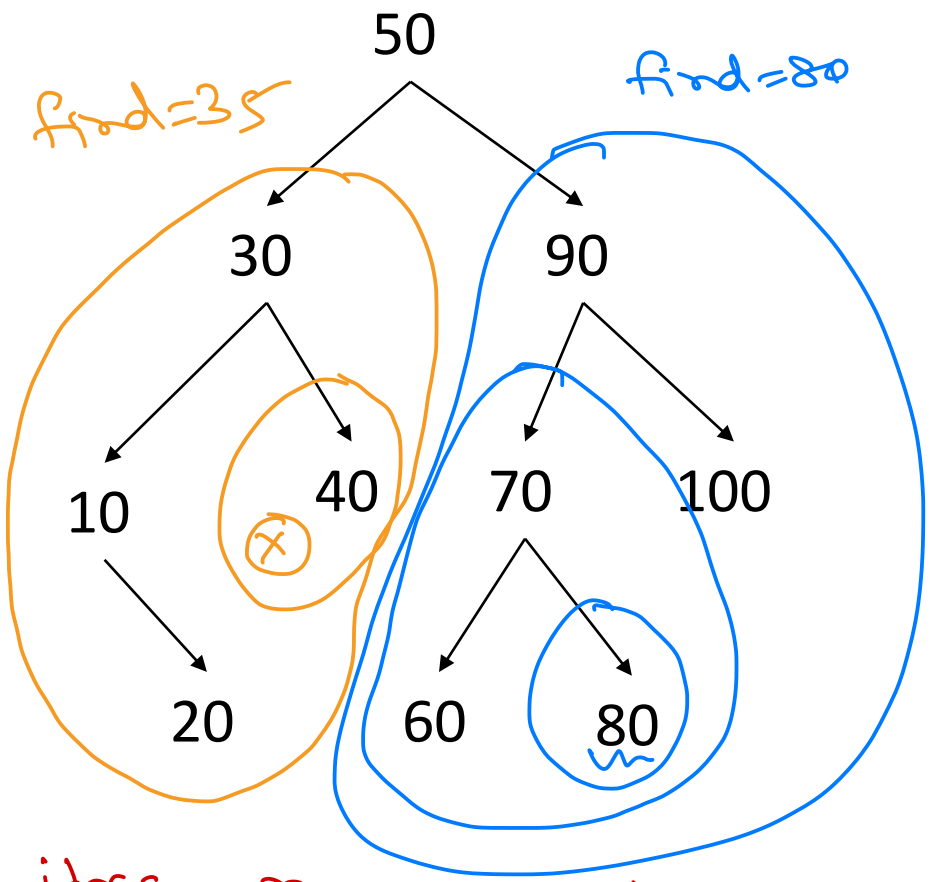
Nilesh Ghule

Recursive add()




```
root
void add (trav, val) {
  if (val < trav.data) {
    if (trav.left == null)
      trav.left = new Node(val);
    else
      add (trav.left, val);
  } else {
    if (trav.right == null)
      trav.right = new Node(val);
    else
      add (trav.right, val);
  }
}
```

BST – search (recursive)



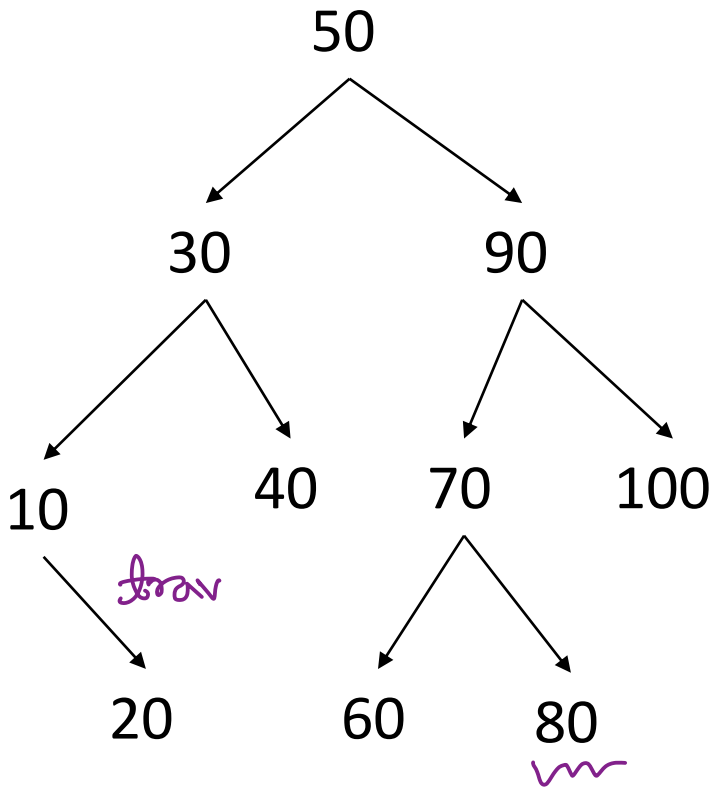
itrs = num of levels

T α h (height)

O(h)

```
Node find(Node trav, int val) {
    if(trav==null)
        return null;
    if(val==trav.data)
        return trav;
    if(val < trav.data)
        t = find(trav.left, val);
    else
        t = find(trav.right, val);
    return t;
}
```

BST – search



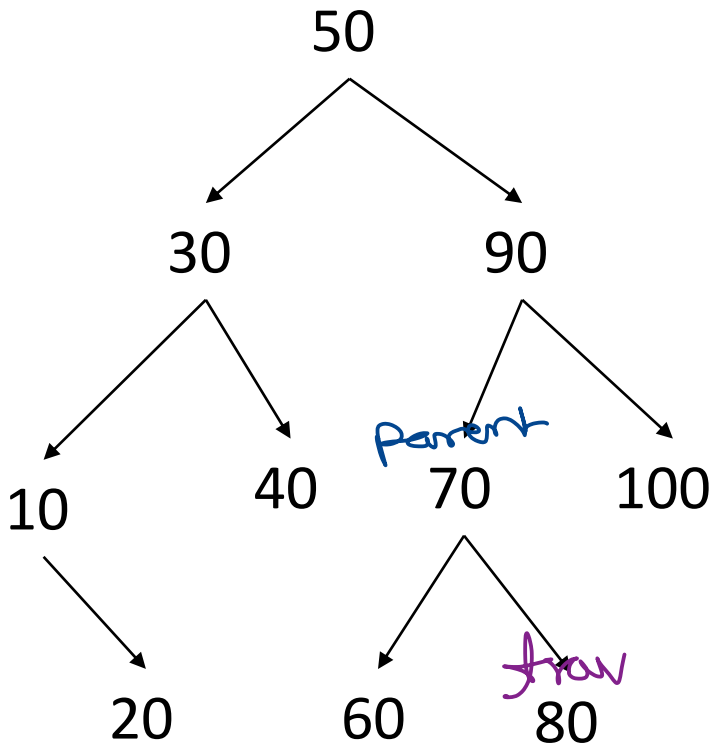
$T \propto O(h)$ ← height

```
trav = root;
while (trav != null)
{
    if(val == trav.data)
        return trav;

    if( val < trav.data)
        trav = trav.left;
    else
        trav = trav.right;
}

return null;
```

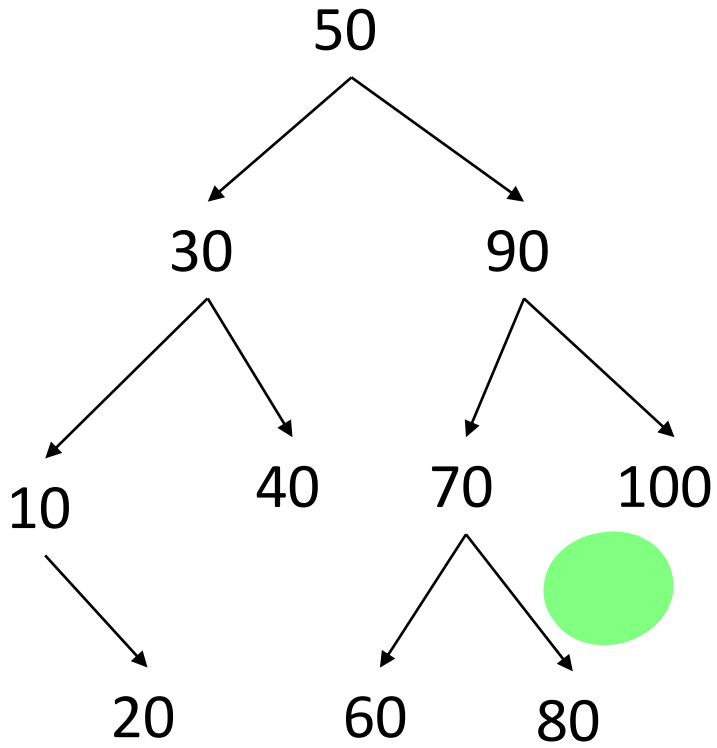
BST – search – with parent



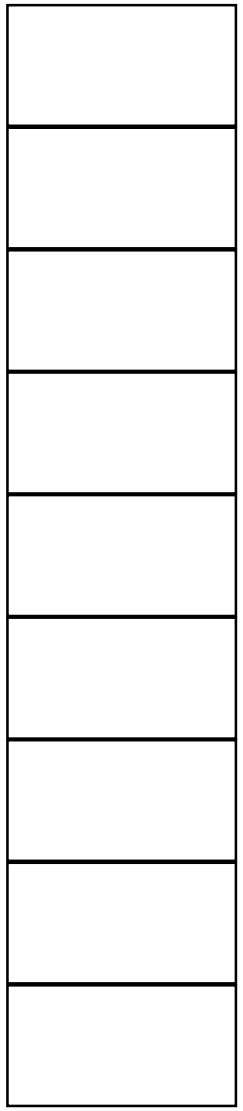

```
Parent = null;
trav = root;
while (trav != null)
{
    if(val == trav.data)
        return trav;
    Parent = trav;
    if( val < trav.data)
        trav = trav.left;
    else
        trav = trav.right;
}
parent = null;
return null;
```

BST – PreOrder

PLR

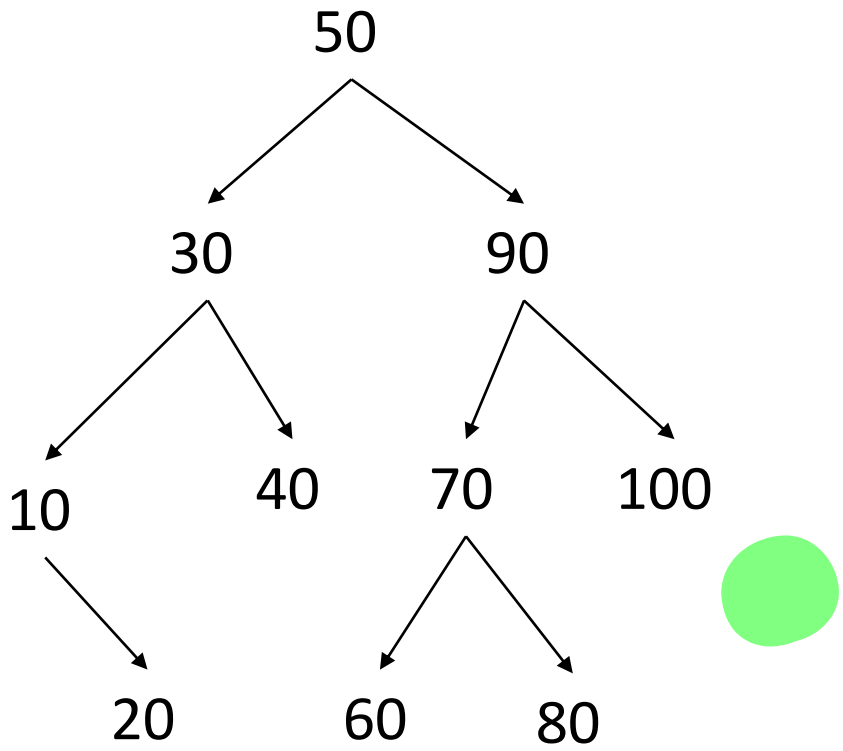


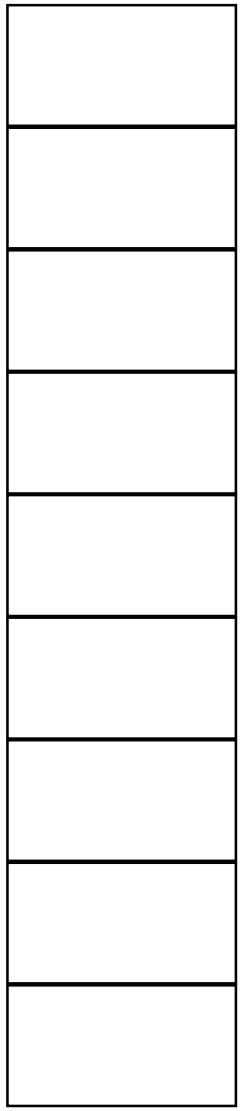
50 30 10 20 40
90 70 60 80 100



```
trav = root;
while(trav != null || !s.isEmpty()){
    while(trav != null){
        print(trav.data);
        if(trav.right != null)
            s.push(trav.right);
        trav = trav.left;
    }
    if(!s.isEmpty())
        trav = s.pop();
}
```

BST – InOrder L P R



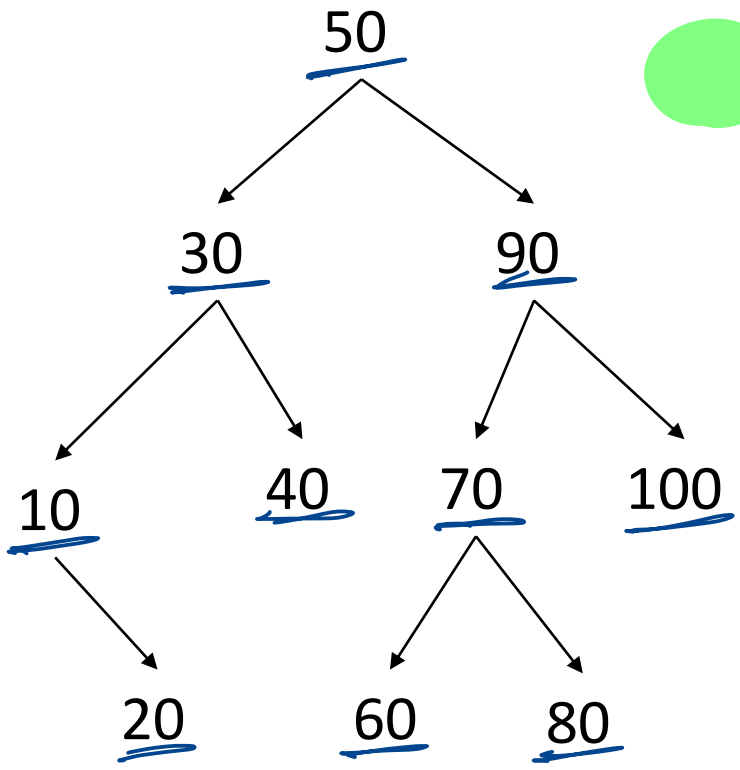


```
trav = root;
while(trav!=null || !s.isEmpty())
{
    while(trav!=null){
        s.push(trav);
        trav = trav.left;
    }
    if(!s.isEmpty()){
        trav = s.pop()
        print(trav.data);
        trav = trav.right;
    }
}
```

10 20 30 40 50

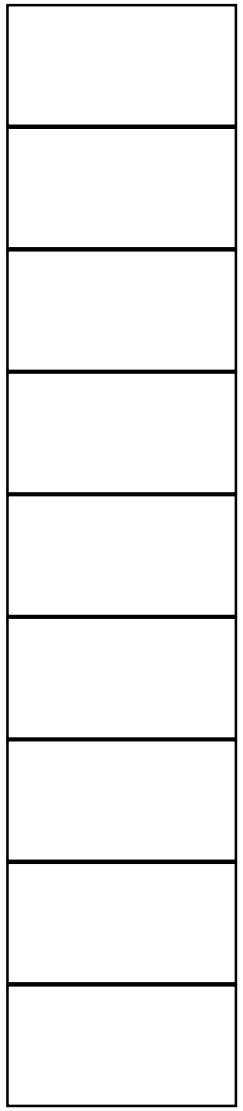
60 70 80 90 100

BST –PostOrder L R P



20 10 40 30 60

80 70 100 90 50



```
trav=root;
while(trav!=null || !s.isEmpty()){
    while(trav!=null){
        s.push(trav);
        trav=trav.left;
    }
    if(!s.isEmpty()){
        trav=s.pop();
        if(trav.right==null||trav.right.visited){
            print(trav.data);
            trav.visited=true;
            trav=null;
        }
        else{
            s.push(trav);
            trav=trav.right;
        }
    }
}
```

Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>