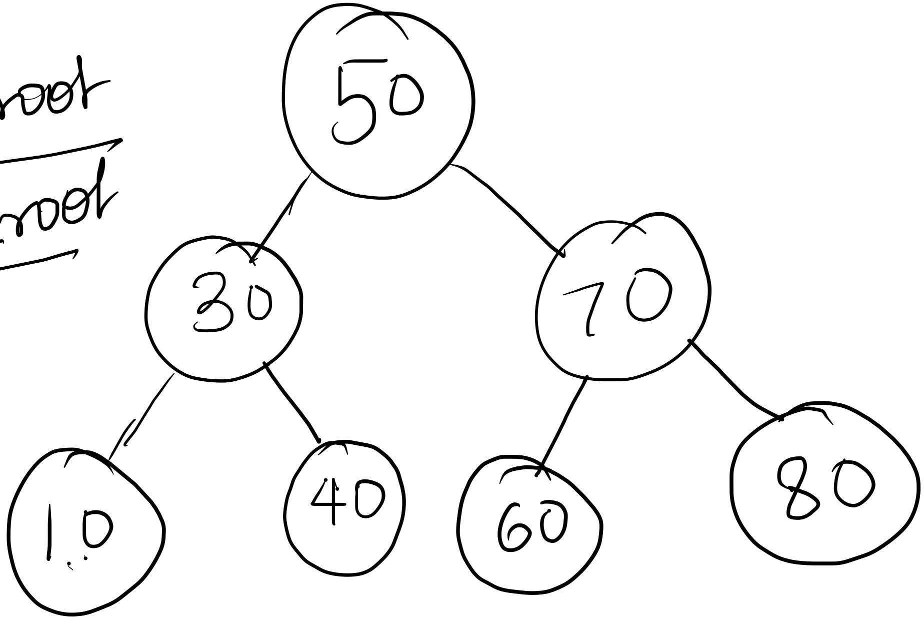
BINARY SEARCH TREE

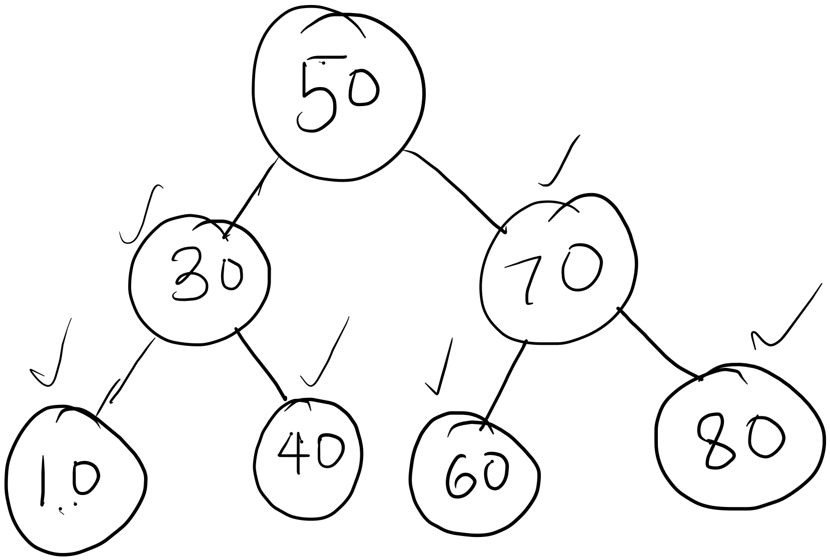
left <= root

right >= root



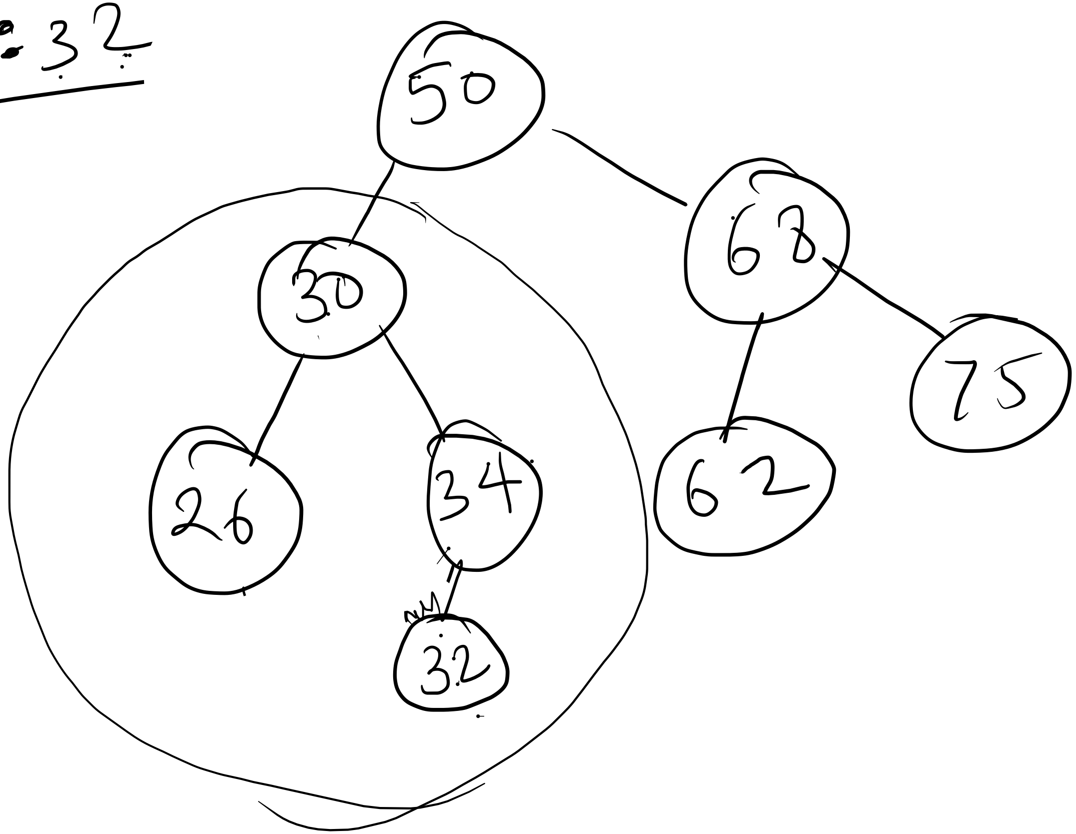
Inorder : 10, 30, 40, 50, 60, 70, 80

→ sorted

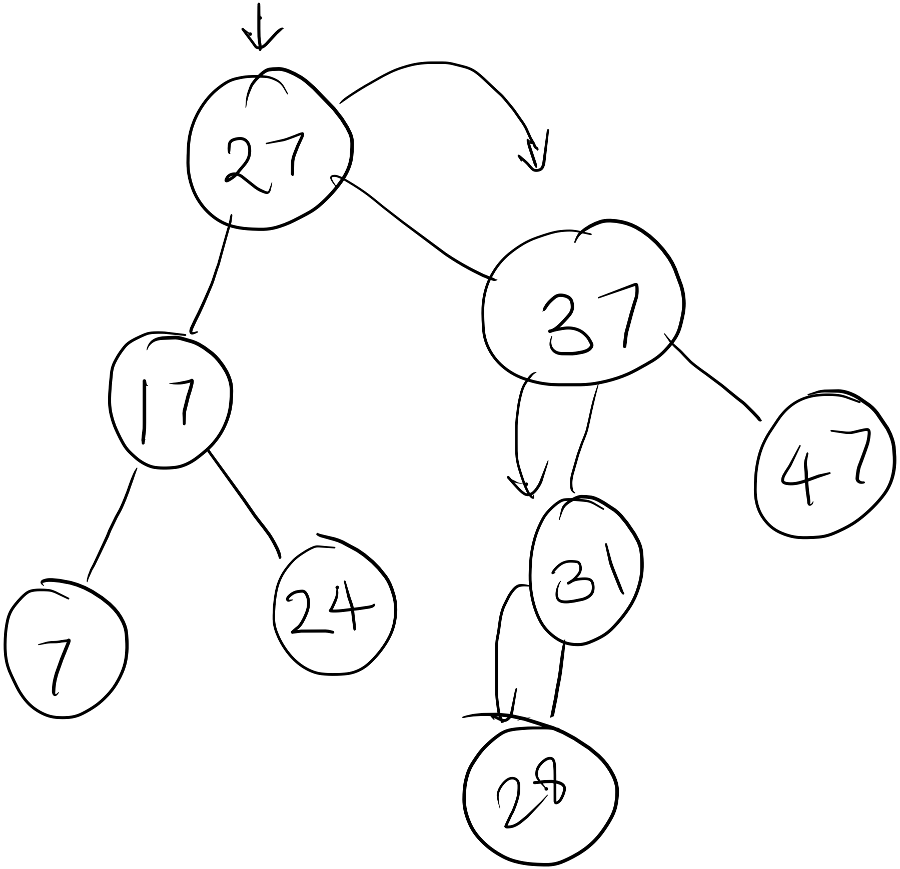


① Insertion in BST

Insert : 32



Insert : 28

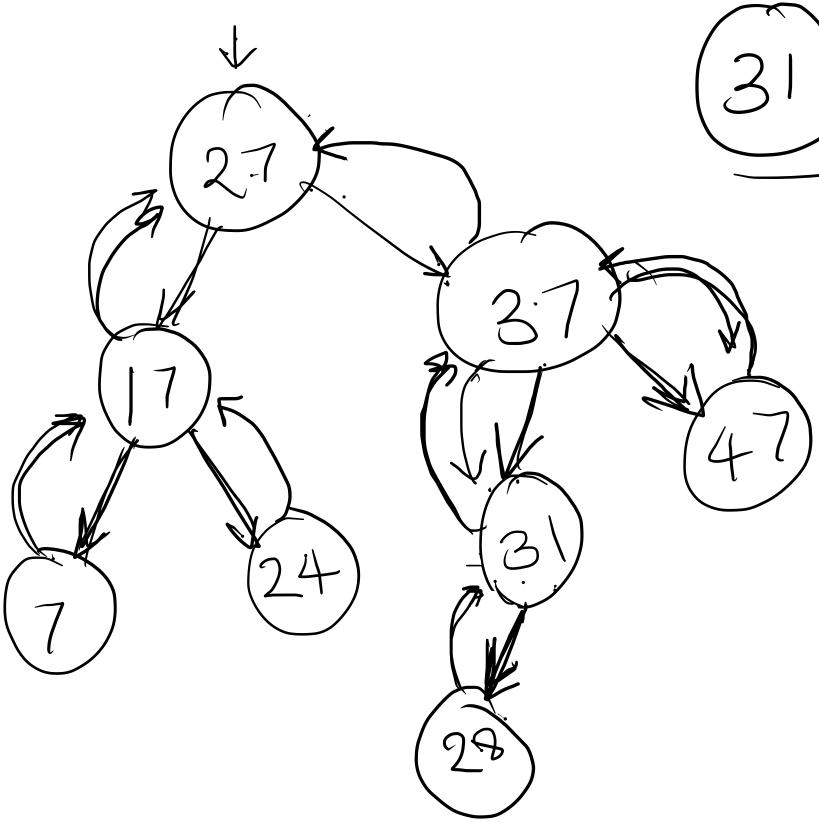


```
if (Key < root.val)
        left

if (Key > root.val)
        right
```

```
if ( root == null )
    → insert (key)
```

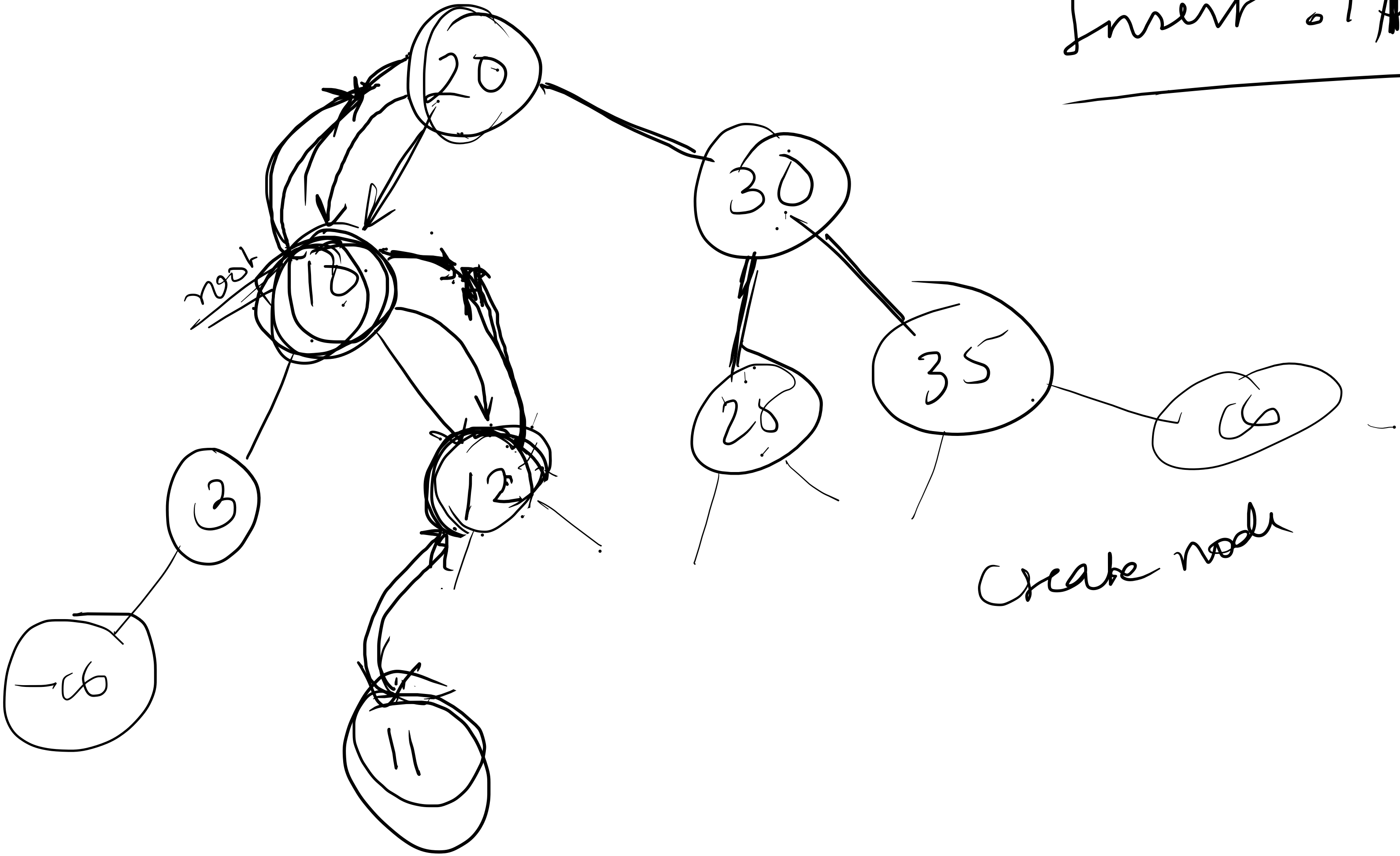
Insert : 28



31 . left = 28

10

Insert :1~~8~~

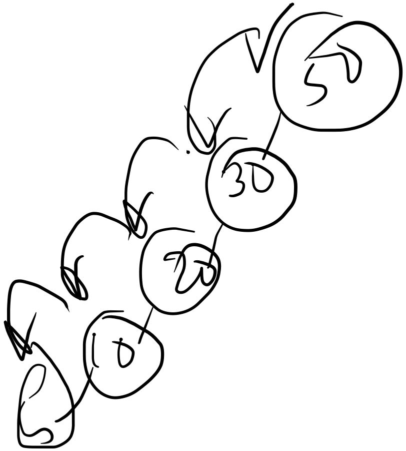


Create node

average case

T.C = $O(\log n)$ $O(\log n)$

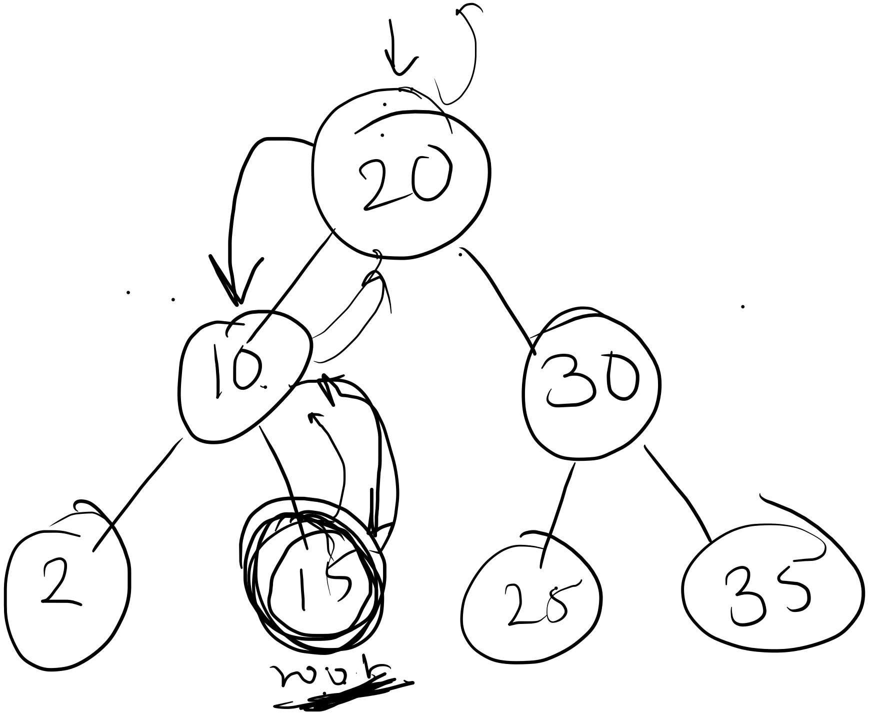
Skewed tree → $O(n)$



$O(n)$

Aux space = $O(n)$

Search in BST



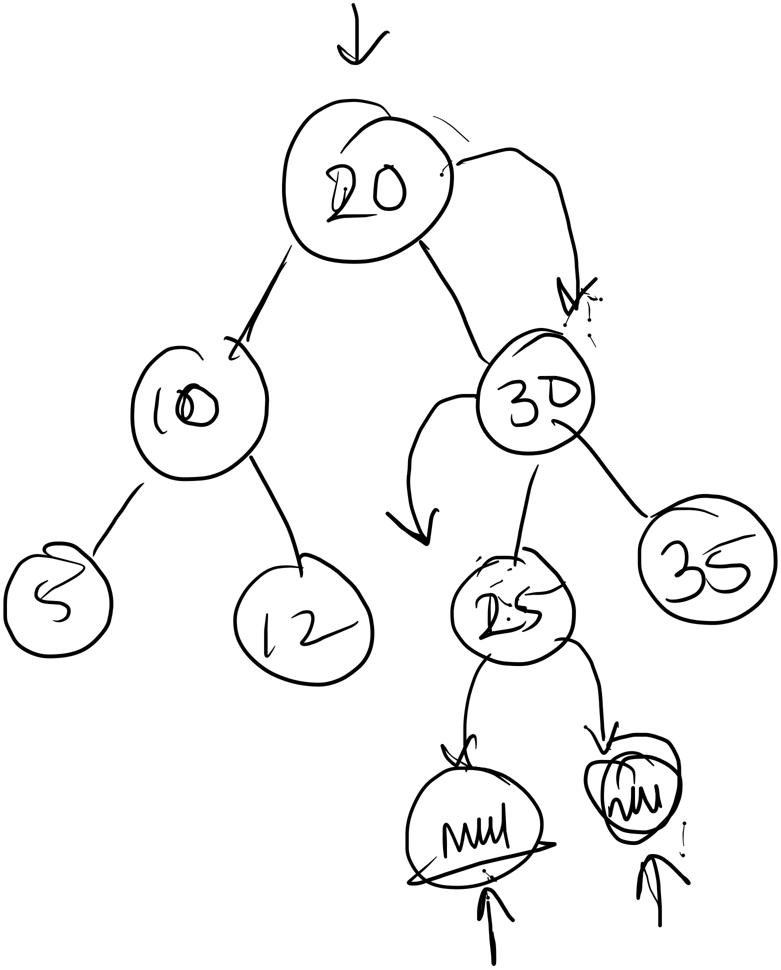
Search: 15

root.val = Key

return root

```
if ( root == null )
    return null
if ( root.val == key )
    return root
if ( key < root.val )
    return search ( left )
else
    return search ( right )
```

key)



Key = 23

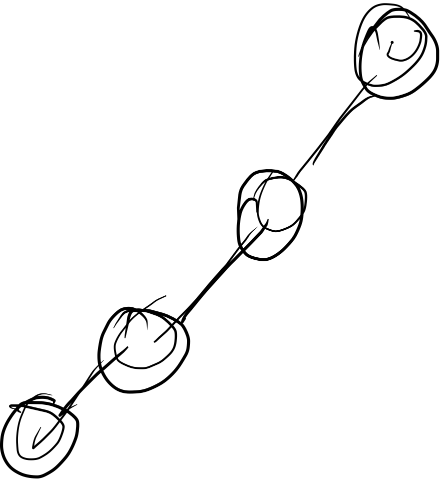
if (root = null)
return null

T.C = $O(logn)$

Average case

SC = $O(logn)$

SC = $O(n)$



Skewed tree = $O(n)$

TC →

TC