1. 1. 
   2. Inorder: 15,10,20,27,30,32,40,50,60,63,70,80

Preorder: 50,20,60,10,40,70,15,30,63,80,27,32

Postorder: 15,10,27,32,30,40,20,63,80,70,60,50

* 1. 

1. 1. struct Node{

int value;

Node\* left;

Node\* right;

Node\* parent;

}

* 1. void insert(int v, Node\* n){

if n is nullptr, set a new node with v as root

if n value is equal to v, return

if v < n.value

if n.left is null:

set new node with value and parent as n then return

else:

call insert(v,n.left)

repeat above for v>n.value, but check for n.right

}

* 1. 
  2. [7,5,6,2,0,4]
  3. [6,5,4,2,0]

1. 1. O(C+S)
   2. O(logC + S)
   3. O(log(S)+log(C))
   4. O(log(S))
   5. O(1)
   6. O(log(C) + S)
   7. O(Slog(S))
   8. O(Clog(S))