19/2/24

Binary search tree

__/__/____

```
#include <stdio.h>
#include <stdlib.h>

Struct Treenode
{
int val;
Struct Treenode * left;
Struct Treenode * right;
};

Struct treenode * create node (int val)
{
Struct Treenode * newnode = (struct tree node *)
malloc( Size of (Struct treenode ));
newnode → val = val;
newnode → left = left;
newnode → right = right;
return newnode;
};
Struct treenode * insert (Struct treenode * root,
int val)
{
if (root == NULL)
{
  return (creatonode (val);
}
```

__/__/____

```
if (val < root -> val)
  {
    root -> left = insert (root -> left, val);
  }
else if ( val > root -> val)
  {
   root -> right = insert (root -> right, val);
  }
 return root.
};
Void inorder (Struct treenode*root)
{
if ( root != NULL)
{
 inorder ( root -> left);
 printf ("%d", root -> val);
 inorder ( root -> right);
}
}
Void postorder. (Strunt treenode * root)
 {
  if (root != NULL)
    {
      postorder (root -> left)
      postorder (root -> right);
```

__/__/____

```
        Printf ("%d", root->val);
    }
}

void preorder (Struct treenode* root)
  {
if (root != NULL)
{
printf ("%d", root->val);
 preorder ( root->left);
  preorder ( root->right);
}
}
Void display (Struct treenode* root)
{
 printf (" Inorder traversal");
  inOrder (root);
 Printf ("\n");
}
   printf (" postorder traversal");
     postorder (root);
     Printf ("\n");

   printf (" preOrder Traversal");
        preorder (root);
              printf ("\n");
}
```

```
int main()
{
Struct treenode* root = NULL;
   root = insert(root, 50);
   insert(root, 30);
   insert(root 20);
   insrt(root 40);
   insrt (root, 70);
    insrt(root, 60);
   insrt (root, 80);
   display(root);
}
```

→ inorder traversal

20 30 40 50 60 70 80

postorder traversal

20 40 30 60 80 70 50

preorder traversal

50 30 20 40 70 60 80