

19-02-24

- ① WAP to construct a Binary Search Tree
- ② Traverse the tree using inorder, postorder, preorder
- ③ Display the elements in tree

```
→ struct Node {  
    int data;  
    struct Node * left;  
    struct Node * right;
```

};

```
struct Node * createNode (int data)  
{
```

```
    struct Node * newNode  
    = (struct Node *) malloc (sizeof (struct Node));
```

```
    newNode->data = data;
```

```
    newNode->left = newNode->right = NULL;
```

```
    return newNode;
```

};

```
struct Node * insert (struct Node * root  
                      int data)
```

```
{
```

```
    if (root == NULL)
```

```
{
```

```
        return createNode (data);
```

```
    }
```

```

if (data < root->data)
{
    root->left = insert (root->left, data);
}
else if (data > root->data)
{
    root->right = insert (root->right, data);
}
return root;
}

```

```

void inorderTraversal (struct Node* root)
{
    if (root != NULL)
    {
        inorderTraversal (root->left);
        printf ("%d", root->data);
        inorderTraversal (root->right);
    }
}

```

```

void postorderTraversal (struct Node* root)
{
    if (root != NULL)
    {
        postorderTraversal (root->left);
        postorderTraversal (root->right);
        printf ("%d", root->data);
    }
}

```

```

void preorderTraversal(struct Node* root)
{
    if (root != NULL)
    {
        printf("%d", root->data);
        preorderTraversal(root->left);
        preorderTraversal(root->right);
    }
}

```

```

void display(struct Node* root)
{
    printf("Elements in the tree:");
    inorderTraversal(root);
    printf("\n");
}

```

output:

Elements in the tree: 20 30 40 50 60 70 80

inorderTraversal: 20 30 40 50 60 70 80

postorderTraversal: 20 40 30 60 80 70 50

preorderTraversal: 50 30 20 40 70 60 80

19.08.24