```
struct node
{ int data;
  struct node * left;
  struct node * right;
}
struct node *root = NULL

struct node * create ( )
{ struct node * temp;
  temp = (struct node*) malloc (size of (struct node));
  temp -> data = new_item;
  temp -> left = temp -> right = NULL
  return temp; }

void insert (nostruct node * root, struct node * temp)
{  if( temp -> data < root -> data)
    {  if(root -> left != NULL)
          insert (root -> left, temp);
       else
          root -> left = temp;
    }
    if (temp -> data > root -> data)
    {  if (root -> right != NULL)
          insert (root -> right, temp);
       else
          root -> right = temp;
    }
}
```

Display methods

```
void inorder (struct node * root)
{  if(root != NULL)
   if (root == NULL)
   { printf (" Tree is empty "); }

   else { if(root != NULL)
          inorder (root -> left);
          printf (" %d", root -> key);
          inorder (root -> right);
        }
}

  void preorder (struct node * root)
{  if (root != NULL)
   { printf ("%d", root -> data);
     preorder ( root -> left);
     preorder (root -> right);
   }
else
  printf (" Tree is Empty ");
}
  void postorder( struct node * root)
{  if (root != NULL)
   { postorder (root -> left);
     postorder (root -> right )
     printf ("%d", root -> data ); }
else
     printf (" Tree is Empty ");
}
```