LAB-10

B.S Tree Implementation

① Node *create()
{
    Node* newnode = (Node*) malloc (sizeof(Node));
    printf("\n Enter Element:"); scanf("%d", &newNode->data);
    newnode->left = newnod->right = NULL;
    return newnode;
}

② Node *insert(Node * Root, Node *newNode)
{
  if(root == NULL)
  {
    root = newNode;
  }
  else
  {
   if(newNode -> data > Root->data)
    {
      if (Root->right == NULL)
       Root->right = newNode;
       else
       insert(Root->right, newNode);
    }
   else
    {
      if (Root->left == NULL)
       Root->left = newNode;
       else
        insert(Root -> left, newNode);
    }
  }
}

④ void PreOrder(Node *Root)
{
if(Root != NULL){
printf("%d", Root->data);
PreOrder(Root->left);
PreOrder(Root->right);
}
}

⑤ void inOrder(Node *Root)
{
if(Root != NULL){
inOrder(Root->left);
printf("%d", Root->data);
inOrder(Root->right);
}
}

⑤ void postOrder(Node *Root)
{
if(Root != NULL){
postOrder(Root->left);
postOrder(Root->right);
printf("%d", Root->data);
}
}

⑥ void display(Node *root, int i)
{
int j;
if(root != NULL)
{
display(root->right, i+1);
for(j=0; j<i; j++)
printf("----");
printf("%d\n", root->data);
display(root->left, i+1);
}
}