02/01/2020

**Experiment No: 22**

**SORT USING BINARY SEARCH TREE**

**AIM:**

Write a program to sort a set of numbers using a binary search tree.

**DATA STRUCTURES USED:**

Tree using Linked List

**ALGORITHM:**

Algorithm Insert()

ptr=root flag = False

1. While ptr != NULL

2. If ITEM <= ptr->DATA

3. ptr1 = ptr

4. ptr = ptr->LC

5. Else if ITEM > ptr->DATA

6. ptr1 = ptr

7. ptr = ptr->RC

8. Else

9. Flag=True

10. print “Item already exists”

11. Endwhile

12. If ptr = NULL

13. new= GetNode(NODE)

14. new->LC = NULL

15. new->RC = NULL

16. new->DATA = ITEM

17. If ptr1->DATA < ITEM

18. ptr1->RC = new

19. if ptr1->DATA>ITEM

20. ptr1->LC = new

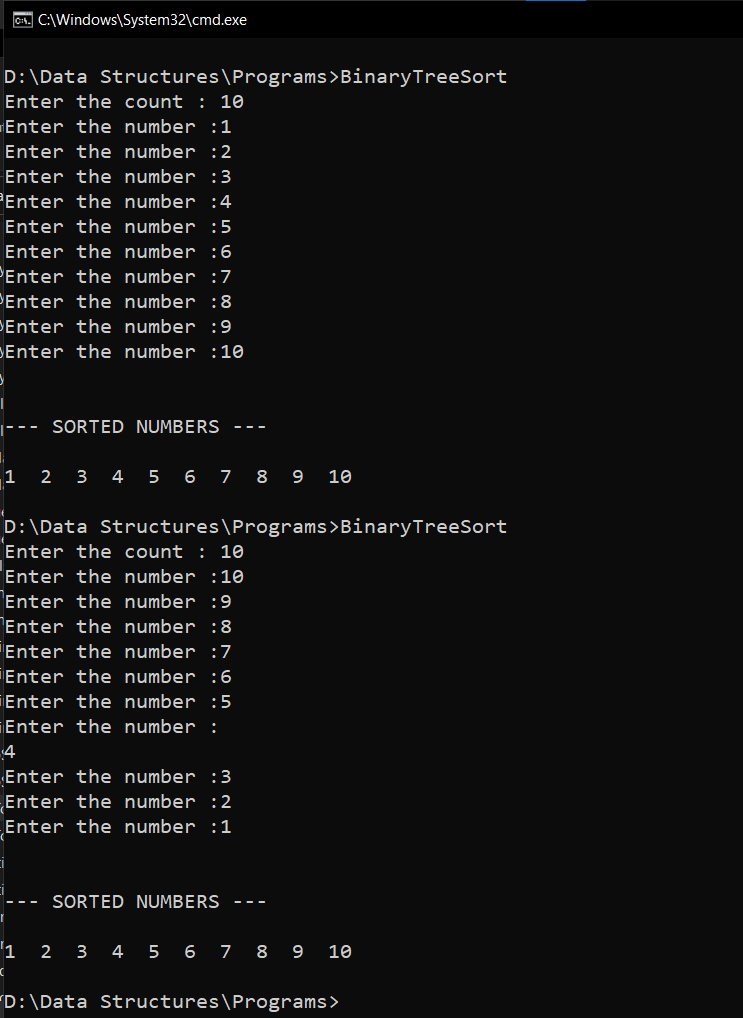
21. Endif

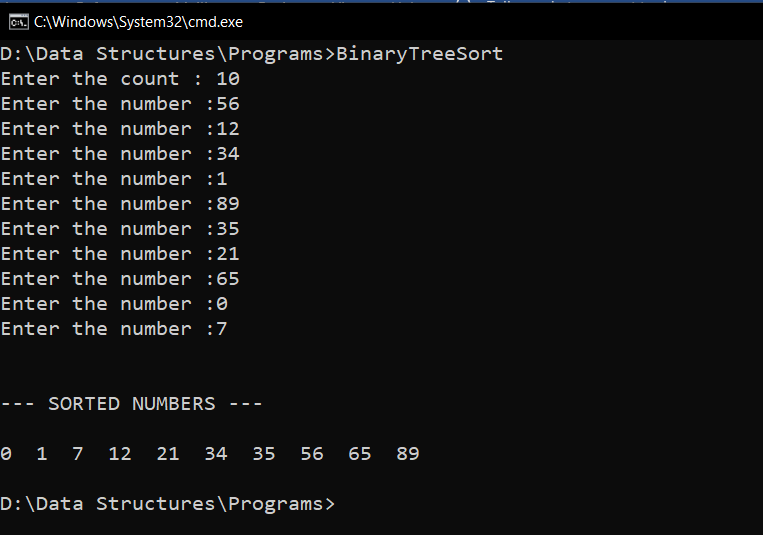
22. EndIf

**PROGRAM:**

#include<stdio.h>  
#include<stdlib.h>  
  
struct node{  
 int data;  
 struct node \*lchild;  
 struct node \*rchild;  
};  
  
  
void Insert(struct node\* root,int item){  
 struct node\* ptr=root;  
 struct node\* ptr1;  
 int flag=0;  
 while(ptr!=NULL && flag == 0){  
 if(item<ptr->data){  
 ptr1=ptr;  
 ptr=ptr->lchild;  
 }else if(item>ptr->data){  
 ptr1=ptr;  
 ptr=ptr->rchild;  
 }else{  
 flag=1;  
 printf("\n ITEM already exists \n ");  
 }  
 }  
 if(ptr==NULL){  
 struct node\* new = (struct node\*)malloc(sizeof(struct node));  
 new->data=item;  
 new->lchild=NULL;  
 new->rchild=NULL;  
 if(ptr1->data<item){  
 ptr1->rchild=new;  
 }  
 if(ptr1->data>item){  
 ptr1->lchild=new;  
 }  
 }  
}  
  
void inorder\_traversal(struct node\* root){  
 struct node\* ptr;  
 ptr = root;  
 if(ptr!=NULL){  
 inorder\_traversal(ptr->lchild);  
 printf("%d ",ptr->data);  
 inorder\_traversal(ptr->rchild);  
 }  
}  
  
  
void main(){  
 int n,i;  
 struct node\* root = (struct node\*)malloc(sizeof(struct node));  
 root->lchild=NULL;  
 root->rchild=NULL;  
 printf("Enter the count : ");  
 scanf("%d", &n);  
 if(n>0){  
 int\* arr = (int\*)malloc(sizeof(int)\*n);  
 for(i=0;i<n;i++){  
 printf("Enter the number :");  
 scanf("%d",&arr[i]);  
 }  
 root->data=arr[0];  
 for(i=1;i<n;i++){  
 Insert(root,arr[i]);  
 }  
 printf("\n\n--- SORTED NUMBERS ---\n\n");  
 inorder\_traversal(root);  
 printf("\n");  
 }else{  
 printf("\n Please Enter valid count ");  
 }  
}

**OUTPUT:**

****

****

**RESULT:**

The given set of numbers are sorted using a binary search tree.