**PUNE INSTITUTE OF COMPUTER TECHNOLOGY DHANKAWADI, PUNE**

**Data Structures And Algorithms(DSA)**

**Assignment No. 06**

**Title : Binary Threaded Tree**

**SE-IT-10**  **ACADEMIC YEAR :- 2020-2021**

**Name :- Diptesh Ravindra Varule Roll No :- 23277**

**Source Code :**

//============================================================================

// Name : dsa\_Assignment6.cpp

// Author : Diptesh Varule

// Version : Updating…..

// Copyright : The Cartel

// Description : Hello World in C++, Ansi-style

//============================================================================

#include<iostream>

#include"TBTREE.h"

int main() {

int choice;

TBT t;

do{

cout<<"\n\*\*\*\*\*\*\*\*MENU\*\*\*\*\*\*\*\*\n1)insert\n2)Preorder traversal"<<endl;

cin>>choice;

switch(choice){

//to insert a new element

case 1:t.create();

t.display\_inorder();

break;

// to display in preorder

case 2:t.preorder();

break;

default:cout<<"Inavlid entry"<<endl;

}

}while(true);

}

**Header File:**

#ifndef TBTREE\_H\_INCLUDED

#define TBTREE\_H\_INCLUDED

#include<iostream>

using namespace std;

struct tnode{

tnode\* llink;

tnode\* rlink;

int data;

bool lthread;

bool rthread;

};

class TBT {

tnode\* head;//root node

public:

TBT();

void create();

void display\_inorder();

void insert();

void preorder();

tnode\* getnode(int);

//virtual ~TBT();

};

TBT::TBT()

{

head=getnode(0);

head->llink=head->rlink=head;

head->rthread=1;

head->lthread=1;

}

tnode\* TBT::getnode(int x){

tnode \*node=new tnode;

node->data=x;//create a node with given data

node->llink=NULL;//set both pointers as NULL

node->rlink=NULL;

node->lthread=node->rthread=0;//set both as 0

return node;

}

void TBT::display\_inorder(){

tnode \*temp=head->llink;//take a pointer and point it to head

while(!(temp->lthread))//traverse till the last left node with no child

temp=temp->llink;

while(temp!=head)

{

cout<<" - "<<temp->data;//print the data of the node

if(temp->rthread)//if no rchild to successor

temp=temp->rlink;

else

{//if rchild present point the temp to it

temp=temp->rlink;

while(!(temp->lthread)&&(temp!=head))//if thatr right child also has further subtree traverse it

temp=temp->llink;

}

}

}

void TBT::preorder(){

tnode \*temp=head->llink;//take a pointer and point it to head

if(temp==head){//if head is null print empty tree

cout<<"Empty tree"<<endl;

return;

}

while(temp!=head){//if head is not null

cout<<" "<<temp->data;//print the data

if(!(temp->lthread)){//if left child is present

temp=temp->llink;//point temp to left child and itrate the llop again

continue;

}

else{//if no left child go to right subtree

while(temp->rthread && temp!=head){//if no subchild

temp=temp->rlink;//point to successor

}

}

if(!(temp->rthread)&&(temp!=head)){//if right child is present

temp=temp->rlink;//point to the next node

}

}

}

void TBT::create(){

tnode \*temp1,\*temp2;//take 2 pointers

int x,choice;//for value and chgoice

bool flag=1;//to check if insertion is done

cout<<"Enter the node value"<<endl;

cin>>x;//accept the value of node

//create root node

if(head->llink==head){//if there is no root node create one

//root node created

temp1=getnode(x);//set 1st pointer to the given data node

temp1->llink=head;//link it to the head

temp1->lthread=temp1->rthread=1;//make both left and right pointer as 1 as no child

temp1->rlink=head;//link right to head as well

head->llink=temp1;//now link head to temp1

head->lthread=0;//and make the lthread as 0 as root node gets created

}

else{//if root node is already present

temp1=head->llink;//point temp1 to root node

do{//keep repeating unitl insertion takes plae

cout<<"curent node :"<<temp1->data<<endl;//print cueenrt data

cout<<"Where to insert new node?\n1)Left Link\n2)Right Link"<<endl;

cin>>choice;//take choice from user where to enter new node

if(choice==1){//for left link

if(temp1->lthread)//check if left subtree is present

{ //this will get executed if no subchild and a node is inserted

temp2=getnode(x);//take 2nd pointer and create node for insertion of new element

temp2->llink=temp1->llink;//pointing to predecessor

temp2->rlink=temp1;//points to successor

temp2->rthread=temp2->lthread=1;//set both threads as 1 as no subchild

temp1->llink=temp2;//now the parent points to the new created node

temp1->lthread=0;//set thread as 0 as new node is created

flag=0;

}

else

temp1=temp1->llink;//if not null go ahead till u find insertion point

}

else{//for right link insertion

if(temp1->rthread){//check if right subtree is present

temp2=getnode(x);//create new node with insertion data with 2nd pointer

temp2->rlink=temp1->rlink;//points to succesor

temp2->llink=temp1;//points to predecessor

temp2->lthread=temp2->rthread=1;//set both as 1

temp1->rlink=temp2;//link parent to child

temp1->rthread=0;

flag=0;

}

else

temp1=temp1->rlink;//if not null go ahead till u find insertion point

}

}while(flag!=0);

}

}

#endif // TBTREE\_H\_INCLUDED

**Output :**

\*\*\*\*\*\*\*\*MENU\*\*\*\*\*\*\*\*

1)insert

2)Preorder traversal

1

Enter the node value

4

- 4

\*\*\*\*\*\*\*\*MENU\*\*\*\*\*\*\*\*

1)insert

2)Preorder traversal

1

Enter the node value

3

curent node :4

Where to insert new node?

1)Left Link

2)Right Link

1

- 3 - 4

\*\*\*\*\*\*\*\*MENU\*\*\*\*\*\*\*\*

1)insert

2)Preorder traversal

1

Enter the node value

6

curent node :4

Where to insert new node?

1)Left Link

2)Right Link

2

- 3 - 4 - 6

\*\*\*\*\*\*\*\*MENU\*\*\*\*\*\*\*\*

1)insert

2)Preorder traversal

1

Enter the node value

8

curent node :4

Where to insert new node?

1)Left Link

2)Right Link

1

curent node :3

Where to insert new node?

1)Left Link

2)Right Link

1

- 8 - 3 - 4 - 6

\*\*\*\*\*\*\*\*MENU\*\*\*\*\*\*\*\*

1)insert

2)Preorder traversal

2

4 3 8 6

\*\*\*\*\*\*\*\*MENU\*\*\*\*\*\*\*\*

1)insert

2)Preorder traversal