

## 实现代码

/\*\*

\* BiTreeOrderCreate.c 二叉排序树的建立了

\*

\* @author darbuly

\* @copyright 2018-2019

\*/

#include <stdio.h>

#include <stdlib.h>

typedef struct BiNode

{

int data;

struct BiNode \*lchild,\*rchild;

}BiNode,\*BiTree;

void insertChild(BiTree \*T,int num);

void CreateBiTreeOrder(BiTree \*T)

{

\*T = (BiNode\*)malloc(sizeof(BiNode));

(\*T)->lchild = NULL;

(\*T)->rchild = NULL;

int num;

scanf("%d",&num);

if((\*T)!=NULL && 65535 != num)

(\*T)->data = num;

while(65535!= num)

{

scanf("%d",&num);

if((\*T)!=NULL && 65535 !=num)

insertChild(T,num);

}

}

void insertChild(BiTree \*T,int num)

{

if(num<(\*T)->data)

{

if((\*T)->lchild == NULL)

{

(\*T)->lchild = (BiNode\*)malloc(sizeof(BiNode));

(\*T)->lchild->lchild=NULL;

(\*T)->lchild->rchild=NULL;

(\*T)->lchild->data = num;

}

else

{

insertChild(&((\*T)->lchild),num);

}

}

else

{

if((\*T)->rchild==NULL)

{

(\*T)->rchild = (BiNode\*)malloc(sizeof(BiNode));

(\*T)->rchild->lchild = NULL;

(\*T)->rchild->rchild = NULL;

(\*T)->rchild->data = num;

}

else

{

insertChild(&((\*T)->rchild),num);

}

}

}

void visit(int c,int level)

{

printf("%d located in %d layer\n",c,level);

}

void MidOrderTraverse(BiTree T,int level)

{

if(T)

{

MidOrderTraverse(T->lchild,level+1);

visit(T->data,level+1);

MidOrderTraverse(T->rchild,level+1);

}

}

int main()

{

printf("BiTreeOrder Test\n");

BiTree T;

int level=1;

CreateBiTreeOrder(&T);

MidOrderTraverse(T,level);

return 0;

}

