二叉树

1.BinaryTree.h文件

#pragma once

#include <stdlib.h>

#include <stdio.h>

typedef struct Node {

int val;

struct Node \*left;

struct Node \*right;

} Node;

void preorder(Node \*root) {

if (root == NULL) {

return;

}

printf("%d ", root->val); // 根

preorder(root->left); // 左子树

preorder(root->right); // 右子树

}

void inorder(Node \*root) {

if (root == NULL) {

return;

}

inorder(root->left); // 左子树

printf("%d ", root->val); // 根

inorder(root->right); // 右子树

}

void postorder(Node \*root) {

if (root == NULL) {

return;

}

postorder(root->left); // 左子树

postorder(root->right); // 右子树

printf("%d ", root->val); // 根

}

Node \*buyNode(int val) {

Node \*node = (Node \*)malloc(sizeof(Node));

node->val = val;

node->left = node->right = NULL;

return node;

}

void Test() {

Node \*a = buyNode(1);

Node \*b = buyNode(2);

Node \*c = buyNode(3);

Node \*d = buyNode(4);

Node \*e = buyNode(5);

Node \*f = buyNode(6);

Node \*g = buyNode(7);

a->left = b;

a->right = c;

c->left = d;

c->right = f;

d->right = e;

f->right = g;

preorder(a); printf("\n");

inorder(a); printf("\n");

postorder(a); printf("\n");

}

1. main.c文件

#include "BinaryTree.h"

int main() {

Test();

}