**实验四实验报告**

**2112066 于成俊**

1. **实验题目**

**给定一个非空二叉树，使用链式存储完成下面操作:**

**1.判断给定二义树是否为完全二叉树**

**2.给定节点值，返回该节点的高度**

**3.若不是完全二叉树，将该树转为完全二叉树并输出。**

**需要自己定义二叉树结构，并根据输入构建二叉树进行上述操作，输入为层次遍历的顺序，若该位置无节点则为#。**

1. **核心代码**

**①节点类 （定义三个指针）**

**char data[3];**

**//用于实现树**

**Node\* left;**

**Node\* right;**

**//用于实现队列**

**Node\* next;**

**②队列类**

**Node\* front1;//用于队列弹出，弹出时往后移**

**Node\* front2;//一直不动，用于从头查找**

**Node\* rear;**

**void pop() {**

**if (front1== NULL)return;**

**front1= front1->next;**

**}**

**③树类**

1. **ListQueue L;//以队列形式存储**

**//用于判断#是否造成了完全二叉树**

**bool t1 ;**

**bool t2 ;**

**Node\* root;**

1. **查找树中左右节点不全的节点**

**//可作为左值的返回值**

**Node\* & find(Node\* n) {**

**if (n->left == NULL)return n->left;**

**else {**

**if (n->right == NULL) return n->right;**

**else find(n->next);**

**}**

**}**

1. **判断树是否为完全二叉树**

**bool judge(Node\* n) {**

**if ( !strcmp(n->data , "#" )&& !t1) {**

**t1 = true; //遇到#记录一次**

**}**

**if (strcmp(n->data, "#") && t1) {**

**t2 = true; //之前遇到了#，且这次不是#**

**}**

**if (t1 && t2) {**

**return false;**

**}**

**if (n->next == NULL)return true;**

**judge(n->next)；}**

**4. void insert(char\* n) {**

**Node\* tmp = new Node(n);**

**if (root == NULL) {**

**root = tmp;**

**L.push(n); }**

**else {**

**find(L.front1) = tmp;**

**L.push(tmp->data);//实际以队列形式存储**

**if (L.front1->right != NULL) {**

**L.pop();//逐层排列关键 }**

**}**

**}**

**5.**

**返回高度**

**int height(char\* c) {**

**int count = 1;**

**Node\* tmp = new Node();**

**tmp = L.front2;**

**while (true) {**

**if (strcmp(tmp->data, c)) {**

**tmp = tmp->next;**

**count++; }**

**else {**

**break; }**

**}**

**int height = (log(count) / log(2)) + 1;**

**return height; }**

1. **算法分析**

**①空间复杂度 一个节点三个指针(6n) 为O(n)**

**②时间复杂度 逐层遍历，类似于顺序查找，为O(n)**

1. **心得体会**
2. **要注意将学到的数据结构都加以利用，不要只想着在一个程序就用一个结构**
3. **可作为左值的返回值进行更改时，是有记忆的，真的更改了**