

树相关

2024年12月30日 星期一 20:37

一、二叉树

1. 层次遍历:

```
void levelOrder(Tree *t) {  
    Queue q;  
    while (t != NULL) {  
        visit(t);  
        if (t->leftChild != NULL) {  
            q.push(t->leftChild);  
        }  
        if (t->rightChild != NULL) {  
            q.push(t->rightChild);  
        }  
        try {t = q.front();}  
        catch (queueEmpty) {return;}  
        q.pop();  
    }  
}
```

二、堆

1. 插入: 把新元素插入新节点, 然后沿从新节点到根节点的路径, 进行一趟冒泡排序
2. 删除: 删掉最后一个节点, 把最后一个节点的数放在根节点, 再从上往下调整顺序
3. 初始化: 从最后一个有孩子的结点开始检查, 确保每个子树都是大/小根堆

三、左高树

1. 定义: 一棵二叉树称为高度优先左高树当且仅当其任何一个内部节点的左孩子的高度都大于或等于右孩子高度

最大(小) HBLT: 一棵HBLT同时还是大(小)根树

四、霍夫曼树

1. 构造过程: 先让每个符号为一个只有一节点的二叉树, 然后不断从集合中选择两棵权最小的二叉树, 把它们合并成新的二叉树直到仅剩下一棵树为止