

高阶程序设计第三次报告

陈德创 19030500217

计算机科学与技术学院

日期：2020 年 4 月 17 日

目录

1	总览	2
2	P1827、美国血统	2
3	P4715、淘汰赛	2
4	P4913、二叉树深度	2
4.1	题目复述	2
4.2	解题思路	2
4.3	复杂度分析	3
4.4	结果	3
5	个人总结	3

1 总览

✓	P4715	【深基16.例1】淘汰赛	普及/提高-
✓	P4913	【深基16.例3】二叉树深度	普及/提高-
✓	P1827	[USACO3.4]美国血统 American Heritage	USACO 普及-

2 P1827、美国血统

这题实际上就是上课讲的拿到题，给出树的前序遍历和中序遍历，求树的后序遍历。



3 P4715、淘汰赛

每轮比赛，假设现在一共有 $2k$ 支队伍，放在数组下标为 $1, 2, \dots, 2k$ ，那么获胜的就有 k 支队伍，可以通过遍历数组来求。那么将这 k 支队伍放在数组下标为 $1, 2, \dots, k$ 的位置，重复操作，直到只剩两支队伍了就好了。



4 P4913、二叉树深度

4.1 题目复述

给出每个节点的两个儿子节点，建立一棵二叉树（根节点为 1），如果是叶子节点，则输入 0 0。建好树后希望知道这棵二叉树的深度。二叉树的深度是指从根节点到叶子结点时，最多经过了几层。

4.2 解题思路

这题其实主要问题在建树。我们用数组模拟链表来建树。先定义一个结构体 `node`，有属性 `l`, `r`，用于表示一个结点，其中 `l` 为左子树根节点的数组下标，`r` 为右子树根节点的数组下标。如果子树为空，则值为 0。再定义一个 `node` 的数组 `a`，`a[i]` 表示这个结点，即第 `i` 个结点，`a[i].r` 是 `i` 结点的右子树根结点。这样我们就容易根据题目数据建树。每个结点题目是给输入左右子树根节点编号的，每次输入的 `x`, `y`，只需令 `a[i].l = x, a[i].r = y` 即可。

树的深度跑一遍 `dfs` 就可以了。函数原型为 `dfs(int x, int dep)`，分别表示当前遍历到第 `i` 个结点，当前深度为 `dep`。边界条件为 `x == 0`，即当 `x` 为 0 时直接返回。对于每个结点，对其左右子树根结点都调用这一函数即可。初始条件为 `dfs(1, 1)`。

4.3 复杂度分析

因为就是跑了一遍 dfs，所以时间复杂度应该为 $O(n)$ ，空间复杂度也是如此。

4.4 结果



5 个人总结

感觉树这一块目前的问题主要还是在如何建树(因为刚开始学),即如何去描述一个树。P4715说明了,即使是两个有一定规则的序列也可以唯一确定一个树,我觉得很有启发性。其实用指针式的链表建树有很多不方便的地方,比如在题目给结点编号的情况下,我们去寻找某一结点也是只能一个一个去找,而不能直接定位。而数组建树就很好了,可以依靠下标访问,而且因为题目中的数据范围是给定的,一般没需要动态分配内存,速度应该也会快一些。