

P4715 【深基16.例1】淘汰赛

题目描述

有 2^n ($n \leq 7$) 个国家参加世界杯决赛圈且进入淘汰赛环节。已经知道各个国家的能力值，且都不相等。能力值高的国家和能力值低的国家踢比赛时高者获胜。1 号国家和 2 号国家踢一场比赛，胜者晋级。3 号国家和 4 号国家也踢一场，胜者晋级……晋级后的国家用相同的方法继续完成赛程，直到决出冠军。给出各个国家的能力值，请问亚军是哪个国家？

输入格式

第一行一个整数 n ，表示一共 2^n 个国家参赛。

第二行 2^n 个整数，第 i 个整数表示编号为 i 的国家的能力值 ($1 \leq i \leq 2^n$)。

数据保证不存在平局。

输出格式

仅一个整数，表示亚军国家的编号。

输入样例

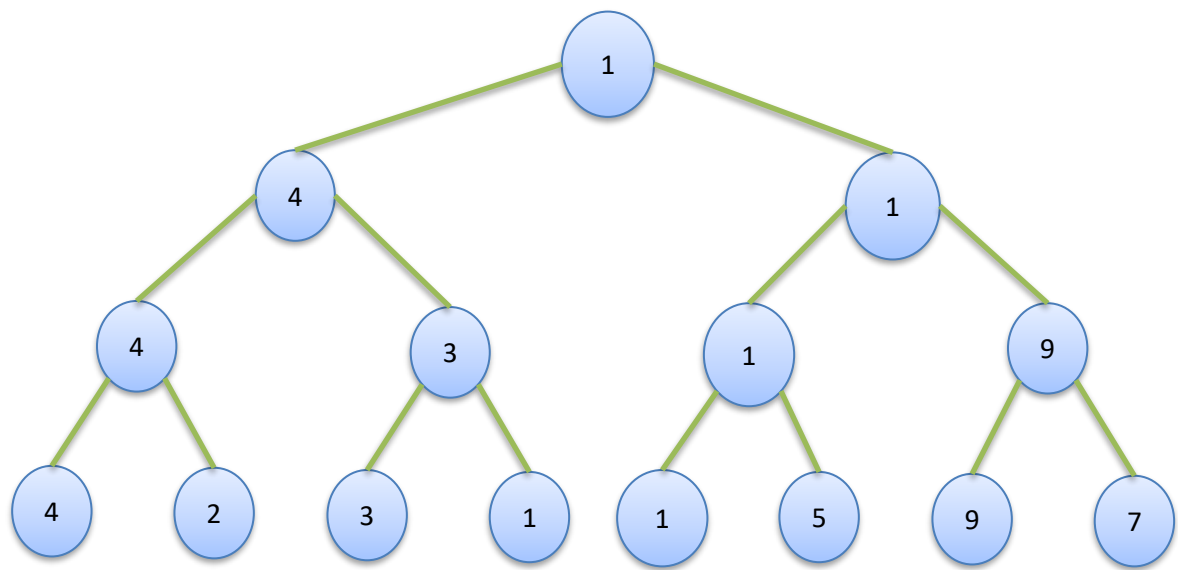
```
3
4 2 3 1 10 5 9 7
```

输出样例

```
1
```

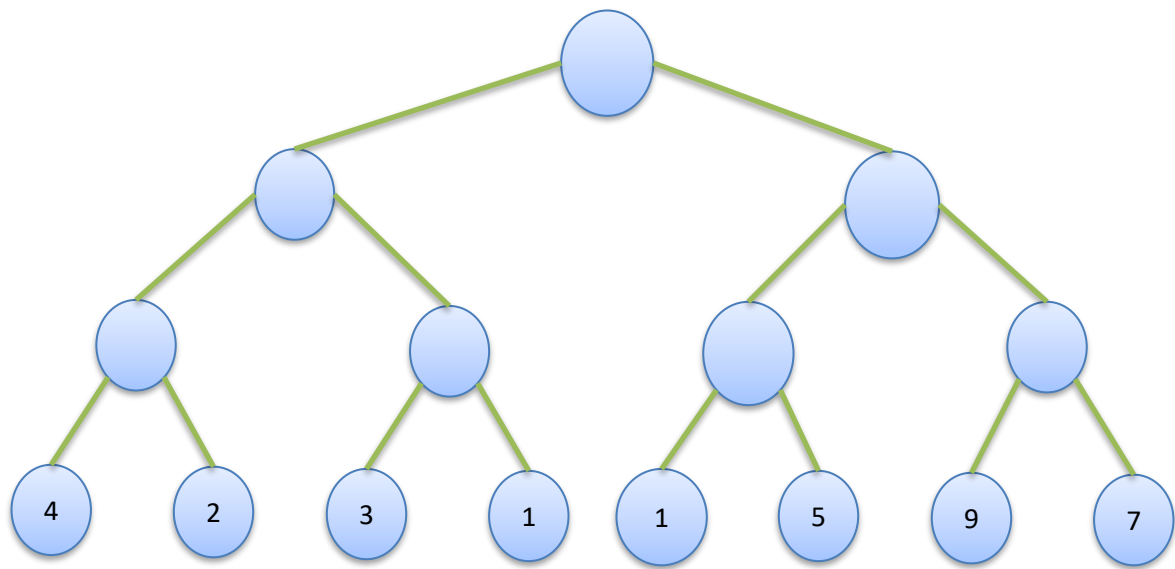
解析

先来绘制一下比赛晋级图像。



通过观察图像，我们发现能力值排名第二的9，并不一定是冠军，因为他很可能一上来就遭遇到了更加强劲的对手。

那么如何才能求出正确的亚军呢？整个图像的初始状态如下：



在最初时刻，我们只知道8个参赛球队的基础能力值，接下来，我们就需要逐级去将晋级的过程计算填写至表中。

在这里，我们使用递归的方法。从根节点(编号为1)开始，依次的向下进行遍历和判断。因为我们当前的二叉树为完全二叉树，就意味着左子树的编号必然是 $2 \times \text{root}$ ，而右子树必然是 $2 \times \text{root} + 1$ 。

根据这样的性质，我们就能够将整个晋级图像填充完毕。

最后，我们再比较第2号结点和3号结点，取二者中较大的一个值即为答案。

编码

```
#include <bits/stdc++.h>

using namespace std;
//2^7是极大值,就是 128是队伍的最大值,完全二叉树的最底层是128,
//上面所有的结点个数肯定是127个,加在一起是255个
//声明一个大于255的数字N可以保存的下
const int N = 260;
int value[N]; //每队伍的能力值
int winner[N]; //胜利者的队伍编号
int n; //n支队伍

//遍历完美二叉树
void dfs(int x) {
    //如果已经完成了所有队伍的比较, 返回
    if (x >= 1 << n) {
        return;
    }
    //左子树,完成左侧value数组与winner数组的填充
    dfs(2 * x);
    //右子树,完成右侧value数组与winner数组的填充
    dfs(2 * x + 1);
    //填充完毕后,可以大胆的使用了~
    int lvalue = value[2 * x];
    int rvalue = value[2 * x + 1];
    //如果左结点获胜
    //记录下获胜方的能力值
    //和获胜方的编号
    if (lvalue > rvalue) {
        value[x] = lvalue;
        winner[x] = winner[2 * x];
    } else {
        value[x] = rvalue;
        winner[x] = winner[2 * x + 1];
    }
}

int main() {
    //2^n次的n,用例是3,表示2^3=8, 8支队伍参加
    cin >> n;
    //读入
    int m = 1 << n; //到底是几支球队?
    for (int i = 0; i < m; i++) {
        //读入各个节点的能力值,为什么要i+m呢? 因为完美二叉树的保存方法决定的。
        cin >> value[i + m];
        //叶子结点的获胜方就是自己国家的编号,相当于初始化数据
        //没有别人比赛之前,获胜者是自己
        winner[i + m] = i + 1;
    }
}
```

```
    }  
    dfs(1); //从根结点开始遍历, 遍历每个节点  
    //找亚军  
    cout << (value[2] > value[3] ? winner[3] : winner[2]);  
    return 0;  
}
```

逻辑航线培优教育，信息学奥赛培训专家。

扫码添加作者获取更多内容。

