

愉快的代码【8.14】

【7.27】我们特意加入了题解系统，也就是说，姐姐也会重新做一遍给你们的题目（包括选做），然后在第二天的题目前给出姐姐自己的代码和注释作为题解或参考

如果觉得自己的代码略为臃肿，可以参考对比一下姐姐的代码；

如果觉得姐姐的代码不如自己的优秀，也可以尽情地嘲讽姐姐～

【7.30】我们特意加入了团队系统，因为感觉到你们有点像是独立学习的样子，比如说姐姐和你们之间有交流，但是你们之间有没有交流呢姐姐就感受不到啦

所以正好在洛谷上发现了一个团队系统，我们可以在这上面布置作业呀（当然姐姐也会继续以 pdf 形式布置作业，你们也还是要以 pdf 形式交作业哈），然后你们就可以在上面看到其它妹妹们的代码呀（包括 AC 代码和还未 AC 的整个过程的代码和分数呀），觉得她们表现不够自己好的话，就可以在群里尽情地嘲笑她们呀～

然后那上面还有一个比赛功能哇，具体形式和我们平时的机考差不多，暑假差不多结束了我们也会有一次期末模拟机考的哈～

如果你们开心的话，你们也可以联合起来给姐姐布置一次平时的作业呀，或者给姐姐安排一次机考呀，你们都是团队的管理员了哈

【8.1】准备给你们留个有趣的团队大作业：给姐姐安排一次机考～

具体时间、题数、难度、知识点待定～

【8.3】经过了某些人性与道德的思考，得出了一个奇怪的想法：

“我今天把代码解决了，明天姐姐的代码还有兴趣看嘛”

那就当天放出来好啦～

同样地：如果觉得姐姐的代码不如自己的优秀，也可以尽情地嘲讽姐姐～

【8.14】暂定 8.20 下午 2.~5. 期末考，考不好要请姐姐吃饭哦

今天的题目：

知识点：树

简单聊一下代码学习的未来规划～

由于憨憨们日渐变懒懒得敲代码

姐姐也懒得带你们学习啦～

那就暂时计划今明天学习树论

16、17学习图论

18、19复习总结（到时要不要姐姐做个ppt再开个会议总结一下暑假代码学习？）

20期末考然后就正式结束啦

剩下的时间就自行复习开学考试的科目叭～

我们的期末考还是要有奖惩机制滴

这样才显得 姐姐之前比赛的奖励不亏 有趣嘛

还记得我们最初60%的目标嘛？

要不600-的憨憨们请姐姐和600+的妹妹们吃饭叭

突然在思考姐姐带了一个学期and一个暑假的学习

吃一顿海底捞是不是基本操作呢

1、<https://www.luogu.com.cn/problem/P1030>

2、<https://www.luogu.com.cn/problem/P4913>

今天的答案：

8.14问题1：

```
/*
    洛谷P1030：求先序排列
    思想：这个之前数据结构还是程序设计应该也做过，我们知道后序遍历最后一个字母就是树根，再根据中序遍历的性质，树根左边的字符串是左子树，右边的字符串是右子树。我们找到根和左右子树了，如何处理左右子树呢？递归一下叭～
    时间复杂度：最坏情况  $O(n)$ 
*/

#include <iostream>
#include <algorithm>
#include <string>
#define ll long long
using namespace std;

void digui(string s1,string s2){ // 中序、后序
    if (s1.length() == 0) return;
    int i=s1.find(s2[s2.length()-1]); // 中序遍历中找到树根
    cout << s1[i]; // 前序遍历：根、左子树、右子树
    digui(s1.substr(0,i),s2.substr(0,i)); // 左子树递归
    digui(s1.substr(i+1),s2.substr(i,s2.length()-i-1)); // 右子树递归
}

int main() {
    string s1,s2;
    cin >> s1 >> s2;
    digui(s1,s2);
}
```

8.14问题2:

```
/*
    洛谷P4913: 二叉树深度
    思想: 建树后跑一遍DFS/BFS即可
    时间复杂度: O(n)
*/

#include <iostream>
#include <algorithm>
#define ll long long
using namespace std;

int l[1000001], r[1000001], ans; // 比结构体更舒服的建树方式。l[i]、r[i]: i的左右子节点

void dfs(int x, int d) { // 根节点、深度
    if (l[x]) dfs(l[x], d+1);
    if (r[x]) dfs(r[x], d+1);
    ans = max(ans, d);
}

int main() {
    int i, j, n;
    cin >> n;
    for (i=1; i<=n; i++) cin >> l[i] >> r[i];
    dfs(1, 1);
    cout << ans << endl;
}
```

Interesting thing:

也许不会再有啦