愉快的代码【8.15】

【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

今天同样是树论～

1、https://www.luogu.com.cn/problem/P1087

2、https://www.luogu.com.cn/problem/P3884

今天的答案：

8.15问题1：

```
/*
    洛谷P1087：FBI树
    思想：树论问题很多都与递归和搜索练习起来哦，比如这个题目就直接告诉你要递归建树啦
    时间复杂度：O(n)
*/

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

void digui(string s) { // 由于是后序遍历，因此先写左右递归再写输出根
    if (s.length() > 1) { // 递归
        digui(s.substr(0,s.length()/2)); // 左字串
        digui(s.substr(s.length()/2)); // 右字串
    }
    if (s == string(s.length(),'0')) cout << 'B'; // 全0字串
    else if (s == string(s.length(),'1')) cout << 'I'; // 全1字串
    else cout << 'F'; // 有0有1
}

int main() {
    int n;
    string s;
    cin >> n >> s;
    digui(s);
}
```

8.15问题2：

```
/*
    洛谷P3884：二叉树问题
    思想：吉林09年省选题，也就那样～首先也是建树，然后从根开始跑一遍DFS/BFS找出树的深度和宽度，两点之间的距离可以转换成图论最短路来做（因此这里用图的邻接矩阵存树），当然由于树的特殊性质（两点之间路径唯一），用搜索（从一个节点开始跑DFS/BFS）也是可以滴～
    时间复杂度：O(n+最短路)，树上DFS/BFS：O(n)，Dijkstra：O(n^2)，Floyd：O(n^3)
*/

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

```cpp
int n,a[101][101],d[101],w[101],ans1=1,ans2=1; // a：邻接矩阵，d[i]：节点i的深度，w[i]：树第i层的宽度
void dfs(int r) {
    for (int i=1; i<=n; i++) {
        if (a[r][i] == 1) { // 树的下行方向边
            d[i] = d[r]+1; // 记录这个点的深度
            w[d[r]+1]++; // 这个点所在层的宽度+1
            ans1 = max(ans1,d[i]); // 更新深度最大值
            ans2 = max(ans2,w[d[r]+1]); // 更新宽度最大值
            dfs(i);
        }
    }
}

int main() {
    int i,j,k,x,y;
    cin >> n;
    memset(a,63,sizeof(a)); // 这里的63约等于1e9,思考一下为什么?
    for (i=0; i<n-1; i++) {
        cin >> x >> y;
        a[x][y] = 1; // 树的下行方向边
        a[y][x] = 2; // 树的上行方向边
    }
    cin >> x >> y;
    d[1] = 1; // 初始化根
    w[1]++;
    dfs(1);
    for (i=1; i<=n; i++) // 这里采用 Floyd（因为代码最短）
        for (j=1; j<=n; j++)
            for (k=1; k<=n; k++) a[j][k] = min(a[j][k],a[j][i]+a[i][k]);
    cout << ans1 << endl << ans2 << endl << a[x][y] << endl;
}
```

Interesting thing:

也许不会再有啦