

模拟试题 1

时间：8:10–11:40

(请选手务必仔细阅读本页内容)

一. 题目概览

中文题目名称	风铃	阅读程序写结果	轰炸	做运动
英文题目名称	mobiles	result	bomb	running
可执行文件名	mobiles	result	bomb	running
输入文件名	mobiles.in	result.in	bomb.in	running.in
输出文件名	mobiles.out	result.out	bomb.out	running.out
每个测试点时限	1 秒	1 秒	2 秒	2 秒
测试点数目	50	10	20	20
每个测试点分值	2	10	5	5
比较方式	全文比较	全文比较	全文比较	全文比较
题目类型	传统	传统	传统	传统
内存限制	256M	256M	512M	512M

二. 提交源程序文件名

对于 C++语言	mobiles.cpp	result.cpp	bomb.cpp	running.cpp
----------	-------------	------------	----------	-------------

三. 编译选项

对于 C++语言	-lm -std=c++14 -O2
----------	--------------------

四. 注意事项：

- 1、文件名（程序名和输入输出文件名）必须使用英文小写。
- 2、C/C++中函数 `main()` 的返回值类型必须是 `int`，程序正常结束时的返回值必须是 `0`。
- 3、全国统一评测时采用的机器配置为：Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz，内存 8GB。上述时限以此配置为准。
- 4、只提供 Linux 格式附加样例文件。
- 5、特别提醒：评测在当前最新公布的 NOI Linux 下进行，各语言的编译器版本以其为准。

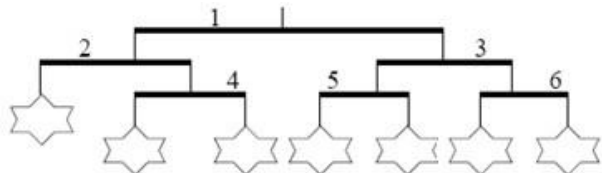
1. 风铃

(mobiles)

【问题描述】

你准备给弟弟 Ike 买一件礼物，但是，Ike 挑选礼物的方式很特别：他只喜欢那些能按照他的特有方式排成有序的东西。

你准备给 Ike 买一个风铃。风铃是一种多层的装饰品，一般挂在天花板上。每个风铃都包含一些由竖直的线连起来的水平杆。每根杆的两端都有线连接，下面或者挂着一根水平杆，或者挂着一个玩具。下面是一个风铃的例子：



为使你的弟弟满意，你需要选一个满足下面两个条件的风铃：

(1) 所有的玩具都在同一层(也就是说，每个玩具到天花板之间的杆的个数是一样的)或至多相差一层。

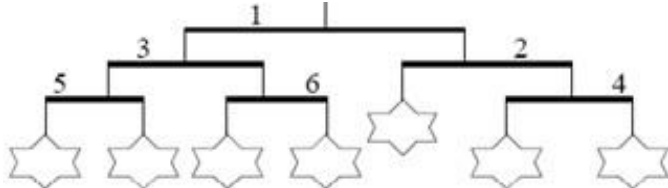
(2) 对于两个相差一层的玩具，左边的玩具比右边的玩具要更靠下一点。

风铃可以按照下面的规则重新排列：任选一根杆，将杆两端的线“交换”。也就是解开一根杆左右两端的线，然后将它们分别绑到杆的另一端。注意这个操作不会改变下面的杆上线的排列顺序。

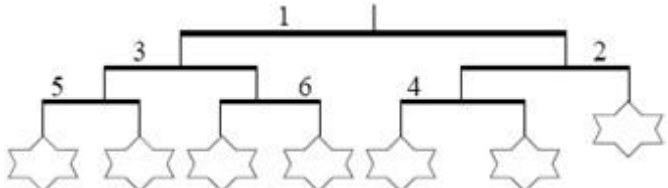
由于你正在参加信息学奥林匹克的训练，所以你决定设计一个算法，判断能否通过重新排列，将一个给定的风铃变为 Ike 喜欢的样子。

考虑上面的例子，上图中的风铃满足条件(1)，却不满足条件(2)——最左边的那个玩具比它右边的要高。但是，我们可以通过下面的步骤把这个风铃变成一个 Ike 喜欢的形式：

1.第一步，将杆 1 的左右两端交换，这使得杆 2 和杆 3 的位置互换，交换的结果如下图所示：



2.第二步，也是最后一步，将杆 2 的左右两端交换，这使得杆 4 到了左边，原来在左边的玩具到了右边，交换的结果如下图所示：



现在这个风铃就满足 Ike 的条件了。你的任务是：给定一个风铃的描述，求出最少需要多少次交换才能使这个风铃满足 Ike 的条件(如果可能的话)。

【输入格式】

输入的第一行包含一个整数 $n(1 \leq n \leq 100000)$ ，表示风铃中有多少根杆。

接下来的 n 行描述杆的连接信息。这部分的第 i 行包含两个由空格分隔的整数 l_i 和 r_i ，描述杆 i 的左右两端悬挂的东西。如果挂的是一个玩具，则对应的值为-1，否则为挂在下面的杆的编号。

如果杆 i 下面挂有其它杆，则这些杆的编号将严格大于 i 。杆 1 位于风铃的顶部。

【输出格式】

输出仅包含一个整数。表示最少需要多少次交换能使风铃满足 Ike 的条件。如果不可能满足，输出-1。

【输入输出样例】

mobiles.in	mobiles.out
6 2 3 -1 4 5 6 -1 -1 -1 -1 -1 -1	2

【样例说明】

上面的样例输入给出了前面描述的示例情形。

【数据范围】

对于 60%的数据，满足 $1 \leq n \leq 20$ ；

对于 100%的数据，满足 $1 \leq n \leq 100000$ ；

2. 阅读程序写结果

(result)

【问题描述】

NOIP2009 普及组初赛阅读程序第三题如下：

```
#include<iostream>
using namespace std;
const int c=2009;
int n,p,s,i,j,t;
int main()
{   cin>>n>>p;
    s=0;
    t=1;
    for(i=1;i<=n;i++)
    {   t=t*p%c;
        for(j=1;j<=i;j++)s=(s+t)%c;
    }
    cout<<s<<endl;
    return 0;
}
```

输入：11 2

输出：_____

这题很简单，谁做都能做出来，但如果扩大 N 的范围，你还能做吗？

【输入格式】

输入 N($1 \leq N \leq 10^4$), P($1 \leq P \leq 1000$), 之间用一个空格隔开。

【输出格式】

输出结果。

【输入输出样例】

result.in	result.out
11 2	782

【数据规模】

对于 50% 的数据， $N \leq 1,000,000$

对于 100% 的数据， $N \leq 10^4$

3. 轰炸

(bomb)

【题目描述】

战狂也在玩《魔方王国》。他只会征兵而不会建城市，因此他决定对小奇的城市进行轰炸。

小奇有 n 座城市，城市之间建立了 m 条有向的地下通道。战狂会发起若干轮轰炸，每轮可以轰炸任意多个城市。

每座城市里都有战狂部署的间谍，在城市遭遇轰炸时，它们会通过地下通道撤离至其它城市。非常不幸的是，在地道里无法得知其它城市是否被轰炸，如果存在两个不同的城市 i, j ，它们在同一轮被轰炸，并且可以通过地道从城市 i 到达城市 j ，那么城市 i 的间谍可能因为撤离到城市 j 而被炸死。为了避免这一情况，战狂不会在同一轮轰炸城市 i 和城市 j 。注意：炸毁的城市还是能够到达的。

你需要求出战狂最少需要多少轮可以对每座城市都进行至少一次轰炸

【输入数据】

第一行两个整数 n, m 。

接下来 m 行每行两个整数 a, b 表示一条从 a 连向 b 的单向边。

【输出数据】

输出一行仅一个整数表示答案。

【样例输入输出】

bomb.in	bomb.out
5 4 1 2 2 3 3 1 4 5	3

【数据范围】

对于 20% 的数据， $n, m \leq 10$ 。

对于 40% 的数据， $n, m \leq 1000$ 。

对于另外 30% 的数据，保证无环。

对于 100% 的数据， $n, m \leq 1000000$ 。

4.做运动

(running)

【问题描述】

一天，Y 君在测量体重的时候惊讶的发现，由于常年坐在电脑前认真学习，她的体重有了突飞猛进的增长。

幸好 Y 君现在退役了，她有大量的时间来做运动，她决定每天从教学楼跑到食堂来减肥。

Y 君将学校中的所有地点编号为 1 到 n，其中她的教学楼被编号为 S，她的食堂被编号为 T，学校中有 m 条连接两个点的双向道路，保证从任意一个点可以通过道路到达学校中的所有点。

然而 Y 君不得不面临一个严峻的问题，就是天气十分炎热，如果 Y 君太热了，她就会中暑。

于是 Y 君调查了学校中每条路的温度 t，及通过一条路所需的时间 c。Y 君在温度为 t 的地方跑单位时间，就会使她的热量增加 t。

由于热量过高 Y 君就会中暑，而且 Y 君也希望在温度较低的路上跑，她希望在经过的所有道路中最高温度最低的前提下，使她到达食堂时的热量最低 (从教学楼出发时，Y 君的热量为 0)。

请你帮助 Y 君设计从教学楼到食堂的路线，以满足她的要求。你只需输出你设计的路线中所有道路的最高温度和 Y 君到达食堂时的热量。

【输入格式】

第一行由一个空格隔开的两个正整数 n,m，代表学校中的地点数和道路数。

接下来 m 行，每行由一个空格隔开的四个整数 a,b,t,c 分别代表双向道路的两个端点，温度和通过所需时间。

最后一行由一个空格隔开的两个正整数 S,T，代表教学楼和食堂的编号。

注意：输入数据量巨大，请使用快速的读入方式。

【输出格式】

输出一行由一个空格隔开的两个整数，分别代表最高温度和热量。

【输入输出样例】

running.in	running.out
5 6	3 24
1 2 1 2	
2 3 2 2	
3 4 3 4	
4 5 3 5	
1 3 4 1	
3 5 3 6	
1 5	

【数据规模与约定】

10%的数据满足 $t=0$

另外 10%的数据满足 $c=0$

另外 30%的数据满足 $n \leq 2000$

100%的数据满足 $n \leq 500000, m \leq 1000000, 0 \leq t \leq 10000, 0 \leq c \leq 10^8, 1 \leq a, b, S, T \leq n, S \neq T$