
CSP-S 模拟赛

竞赛时间：2019 年 11 月 11 日 8:00-11:30

题目名称	骰子的游戏	购物	随机树
可执行文件名	sides	shopping	rand
输入文件名	sides.in	shopping.in	rand.in
输出文件名	sides.out	shopping.out	rand.out
每个测试点时限	1 秒	1 秒	1 秒
内存限制	128 MB	128 MB	256 MB
题目分值	100	100	100
题目类型	传统	传统	传统

提交源程序须加后缀

对于 C++ 语言	sides.cpp	shopping.cpp	rand.cpp
-----------	-----------	--------------	----------

- 说明：1.若无特殊说明，所有评测均过滤行末空格与文末回车；
- 2.需要建立子目录，不要提交多余文件，程序名一律使用小写英文字母；
- 3.编译器以评测机配置为准，评测系统为 NOI Linux；

骰子的游戏(sides.cpp)

【题目描述】

在 Alice 和 Bob 面前的是两个骰子，上面分别写了六个数字。

Alice 和 Bob 轮流丢掷骰子，Alice 选择第一个骰子，而 Bob 选择第二个，如果谁投掷出的数更大，谁就可以获胜。

现在给定这两个骰子上的 6 个数字，你需要回答是 Alice 获胜几率更大，还是 Bob 获胜几率更大。（请注意获胜几率相同的情况）

【输入格式】

输入文件名为 sides.in。

第一行一个数 T，表示数据个数。

接下来的每一组数据一共有 2 行，每一行有 6 个正整数，第一行是第一个骰子上的 6 个数，第二行是第二个骰子上的 6 个数。

【输出格式】

输出文件名为 sides.out。

T 行，每行一个字符串。

如果 Alice 获胜几率更大，你需要输出 Alice；如果 Bob 获胜几率更大，你需要输出 Bob；如果获胜几率一样大，你需要输出 Tie。

【样例输入】

```
2
3 3 3 3 3 3
1 1 4 4 4 4
1 2 3 4 5 6
6 5 4 3 2 1
```

【样例输出】

```
Bob
Tie
```

【样例解释】

第一个数据中，Alice 有三分之一几率获胜，Bob 有三分之二几率获胜；

第二个数据中，Alice 和 Bob 的骰子完全一致，所以获胜几率一样大。

【数据规模与约定】

对于 30% 的数据， $1 \leq T \leq 10$ 。

对于 60% 的数据， $1 \leq T \leq 1000$ 。

对于 100% 的数据， $1 \leq T \leq 10^5$ ，所有输入的数均 $\leq 10^7$ 。

购物(shopping.cpp)

【题目描述】

在遥远的东方，有一家糖果专卖店。

这家糖果店将会在每天出售一些糖果，它每天都会生产出 m 个糖果，第 i 天的第 j 个糖果价格为 $C[i][j]$ 元。

现在的你想要在接下来的 n 天去糖果店进行选购，你每天可以买多个糖果，也可以选择不买糖果，但是最多买 m 个。（因为最多只生产 m 个）买来糖果以后，你可以选择吃掉糖果或者留着之后再吃。糖果不会过期，你需要保证这 n 天中每天你都能吃到至少一个糖果。

这家店的老板看你经常去光顾这家店，感到非常生气。（因为他不能好好睡觉了）于是他会额外的要求你支付点钱。具体来说，你在某一天购买了 k 个糖果，那么你在这一天需要额外支付 k^2 的费用。

那么问题来了，你最少需要多少钱才能达成自己的目的呢？

【输入格式】

输入文件名为 shopping.in。

第一行两个正整数 n 和 m ，分别表示天数以及糖果店每天生产的糖果数量。

接下来 n 行（第 2 行到第 $n+1$ 行），每行 m 个正整数，第 $x+1$ 行的第 y 个正整数表示第 x 天的第 y 个糖果的费用。

【输出格式】

输出文件名为 shopping.out。

输出只有一个正整数，表示你需要支付的最小费用。

【样例输入 1】

```
3 2
1 1
100 100
10000 10000
```

【样例输出 1】

```
107
```

【样例输入 2】

```
5 5
1 2 3 4 5
2 3 4 5 1
3 4 5 1 2
4 5 1 2 3
5 1 2 3 4
```

【样例输出 2】

```
10
```

【数据规模与约定】

对于前 20%的数据, $1 \leq n, m \leq 3$ 。
对于前 40%的数据, $1 \leq n, m \leq 8$ 。
对于前 50%的数据, $1 \leq n, m \leq 10$ 。
对于另 10% (不与前 50%相交) 的数据, $1 \leq n, m \leq 75$, m 个糖果的价格每天都不变。
对于前 70%的数据, $1 \leq n, m \leq 75$ 。
对于 100%的数据, $1 \leq n, m \leq 300$, 所有输入的数均 $\leq 10^6$ 。

随机树(rand.cpp)

【题目描述】

平日里写 hash 的时候，总有某些选手由于脸黑而导致惨遭卡模数，然后一些恶意卡模数的出题人也因此身败名裂。为了防止被卡，我们用一种高级的随机方式来代替原来的线性随机生成，也就是所谓的随机树！

现在有一棵编号为 $0 \sim n-1$ 的有根树，其中 0 是树的根。每个节点初始有一个值 T_i 。现在要求支持一下两种操作：

1. 给出两个正整数 u 和 x ，我们将 T_u 的值乘以 x ，我们将这种操作称为 SEED 操作。
2. 给出一个正整数 i ，询问 S_i 以及它一共有多少个正约数。其中 S_i 表示以 i 为根的子树所有点的权值的乘积，我们将这种操作称为 RAND 操作。

容易发现，这样得到的答案还是很随机的。（其实不是）

你需要回答每一次的询问，由于一个数的约数个数可能非常多，这个数也可以非常大，你只需要把答案对 $1e9+7$ 取模就可以了。

【输入格式】

输入文件名为 rand.in。

第一行一个正整数 n ，表示节点个数。

接下来 $n-1$ 行，每行两个正整数 u 和 v ，表示 u 是 v 的父节点。

接下来一行 n 个正整数，分别表示每个节点的初始权值 T_i 。

接下来一行一个正整数 q ，表示操作的个数。

接下来 q 行，每行是以下两种情况之一：

1. SEED u x

表示将 u 节点的权值乘以 x 。

2. RAND i

表示询问 S_i 以及它一共有多少个正约数。

【输出格式】

输出文件名为 rand.out。

每一行两个整数，对应一个 RAND 操作，你需要输出所求的权值以及它的正约数个数，答案对于 $1e9+7$ 取模即可。

【样例输入】

```
8
0 1
0 2
1 3
2 4
2 5
3 6
3 7
7 3 10 8 12 14 40 15
3
RAND 1
```

SEED 1 13

RAND 1

【样例输出】

14400 63

187200 126

【数据规模与约定】

对于 20%的数据, $1 \leq n, q \leq 10$ 。

对于 40%的数据, $1 \leq n, q \leq 100$ 。

对于 60%的数据, $1 \leq n, q \leq 2000$ 。

对于 80%的数据, $1 \leq n, q \leq 50000$ 。

对于 100%的数据, $1 \leq n, q \leq 100000$ 。

另外请注意, 所有读入的数一定满足 $1 \leq x \leq 10^9$ 。

同时, 数据保证在任意时刻, 每个点的权值不可能拥有超过 13 的素因子, 也就是说, 每个数的素因子最多只有 2, 3, 5, 7, 11, 13 这六种可能。