



NOIP 模拟赛

2018 年 11 月 4 日

题目名称	小迟的比赛	Yuno like cake	格子填数
可执行文件名	game	cake	grid
输入文件名	game.in	cake.in	grid.in
输出文件名	game.out	cake.out	grid.out
每个测试点时限	3 秒	1 秒	1 秒
内存限制	512MB	512MB	512MB
测试点数目	10	10	10
每个测试点分值	10	10	10
是否有 Special Judge	无	无	无
题目类型	传统型	传统型	传统型
是否有附加文件	否	否	否
C++ 语言文件名后缀	cpp	cpp	cpp
C 语言文件名后缀	c	c	c
Pascal 语言文件名后缀	pas	pas	pas

编译开关

对于 C++ 语言	-lm
对于 C 语言	-lm

考试时间 3.5h。

为梦想灼伤了自己
也不要平庸的喘息
我要的并不在这里
你给的答案没意义
此刻我怎么可以输给你
所以我每一个都选 C
就用最轻轻松松的一笔
毁掉你所有的问题



清北学堂®

人生需要规划 高中更应如此 —赵英俊《都选 C》

清北i学堂®

小迟的比赛

`game.in/.out/.cpp`

【问题描述】

小迟最近去参加了一个锦标赛，这个锦标赛总共有 n 轮比赛，最终成绩由这 n 轮比赛中赢的轮数决定。至于小迟每一轮比赛的胜利概率，则取决于他在该轮比赛之前的战绩。也就是说，如果小迟在第 i 轮比赛选择积极应战，并且前 $i-1$ 轮比赛中取得了 j 胜的话，那么第 i 轮比赛的胜率概率为 $p[i][j]$ ，这里我们保证了一点就是对于同一个 i ， $p[i][j]$ 关于 j 的上升保持单调不上升（也就是说 $p[i][j] \geq p[i][j+1]$ ）。

小迟观察到这个规则之后，想到了一个可能可以使他最终成绩更优的方法，就是在某些轮比赛采取第二种策略，故意求败，也就是以 100% 的概率输掉该轮比赛，从而使自己在后面能够遇到更容易对付的对手。

小迟现在已经看到了整个 p 数组，小迟希望你能告诉他一个最优的策略，使得他能最大化他的期望赢的轮数。这里，定义一下期望。假如我们要求一个事件 A 的期望，那么假如事件 A 以 P_i 的概率结果为 i ，那么事件 A 的期望则是 $i \cdot P_i$ 的和，大概的含义就是结果值关于概率的一个加权平均数。

【输入格式】

输入文件名为 `game.in`。

输入数据第一行为轮数 n ， n 为正整数。

接下来的 n 行，第 i 行有 i 个实数，表示对应的 $p[i][0], \dots, p[i][i-1]$ 。

【输出格式】

输出文件名为 `game.out`。

一行一个实数，表示最优策略下期望赢的轮数，保留两位小数。

【样例输入】

0.5

0.5 0.5

【样例输出】

1.00

【样例解释】

由于我们看到对于第 i 轮，无论之前战绩如何，胜率都是相同的，因此，我们的最优策略应当是每一轮努力求胜。

然后，第一轮，如果我们赢了，概率为 0.5，输了的概率也为 0.5。

如果第一轮赢了，第二轮又赢了，概率为 $0.5*0.5=0.25$ ，赢两盘；

如果第一轮赢了，第二轮输了，概率为 $0.5*(1-0.5)=0.25$ ，赢一盘；

如果第一轮输了，第二轮赢了，概率为 $(1-0.5)*0.5=0.25$ ，赢一盘；

如果两轮都输了，概率为 $(1-0.5)*(1-0.5)=0.25$ ，赢零盘。

故期望赢的轮数为 $0.25*2+(0.25+0.25)*1+0.25*0=1$ 。

【数据规模及约定】

对于 30% 的数据， $n \leq 2$ 。

对于 100% 的数据， $1 \leq n \leq 1000, 0 \leq p[i][j] \leq 1$ 。



Yuno like cake

cake.in/.out/.cpp

【问题描述】

双十一就要来啦! 而 Yuno 刚刚获得了一笔 X 元的奖金。那么是不是应该清空下购物车呢?

购物车总共有 N 个物品, 每个物品的价格为 V_i , Yuno 想尽可能地把手头的奖金给花光, 所以她要精心选择一些商品, 使得其价格总和最接近但又不会超过奖金的金额。那么 Yuno 最后最少可以剩下多少钱呢?

【输入格式】

第一行, 两个正整数 N, X 。

第二行, N 个正整数 V_i 表示第 i 个物品的价格。

【输出格式】

输出一个整数, 表示 Yuno 最后最少可以剩下的钱数。

【样例输入 1】

```
4 50
1 2 3 4
```

【样例输出 1】

```
40
```

【样例输入 2】

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



【样例输出 2】

0

【数据规模及约定】

10% 的数据: $N \leq 10$

40% 的数据: $N \leq 20, X, V_i \leq 10000$

100% 的数据: $N \leq 40, X, V_i \leq 10^9$

人生需要规划 高中更应如此



格子填数

`grid.in/.out/.cpp`

【背景】

众所周知，有时候生成数据比解题还难。现在有这么一道题：
一个 $h \times w$ 大小的矩阵（行编号从上往下依次为 $1 \sim h$ ，列编号从左往右依次为 $1 \sim w$ ，第 i 行第 j 列的格子坐标为 (i, j) ），每个格子上都有一个数（数的范围 $1 \sim m$ ）。 n 次询问，每次询问一个子矩阵内数的最大值。

【问题描述】

现在轮到 Kano 给这题造数据了，但他又不会写这题的标程，于是对于每个询问他都先随机出一个答案，接着想通过答案去构造出一个满足所有答案的数据。换句话说，对于构造出来的矩阵，对于每次询问的子矩阵，其中的最大值需要等于 Kano 预先设定的答案。

现在 Kano 已经预先设定好答案了，那么满足要求的矩阵到底有多少个呢？

【输入格式】

第一行为 4 个数 h, w, m, n 。

接下来 n 行，每一行 5 个整数 x_1, y_1, x_2, y_2, v 用来描述一次询问和 Kano 预先设下的答案，表示询问左上角为 (x_1, y_1) 右下角为 (x_2, y_2) 的子矩阵内的格子的最大值，以及预先设下的答案 v 。保证 $1 \leq x_1 \leq x_2 \leq h, 1 \leq y_1 \leq y_2 \leq w$

【输出格式】

输出一行一个整数，表示答案对 $10^9 + 7$ 取模的值。

【样例输入】

```
2 3 2 1
```



1 2 2 3 2

【样例输出】

60

【数据规模和约定】

对 20% 的数据: $h, w, n, m \leq 3$

对另外 10% 的数据: $n = 0$

对另外 20% 的数据: $n = 1$

对 100% 的数据: $1 \leq h, w, m \leq 10000, 1 \leq v \leq m, 0 \leq n \leq 10$

人生需要规划 高中更应如此



(完)