

IOI 2019 中国国家队选拔

CTS 2019

第一试

时间：2019 年 5 月 13 日 08:30 ~ 13:30

题目名称	随机立方体	珍珠	无处安放
题目类型	传统型	传统型	提交答案型
目录	cube	pearl	nowhere
可执行文件名	cube	pearl	N/A
输入文件名	cube.in	pearl.in	nowhere*.in
输出文件名	cube.out	pearl.out	nowhere*.out
每个测试点时限	12.0 秒	1.0 秒	N/A
内存限制	512 MB	512 MB	N/A
子任务数目	10	25	10
测试点是否等分	是	是	是

提交源程序文件名

对于 C++ 语言	cube.cpp	pearl.cpp	N/A
对于 C 语言	cube.c	pearl.c	N/A
对于 Pascal 语言	cube.pas	pearl.pas	N/A

编译选项

对于 C++ 语言	-O2 -std=c++11	-O2 -std=c++11	N/A
对于 C 语言	-O2 -std=c11	-O2 -std=c11	N/A
对于 Pascal 语言	-O2	-O2	N/A

随机立方体 (cube)

【题目描述】

有一个 $n * m * l$ 的立方体，立方体中每个格子上都有一个数，如果某个格子上的数比三维坐标至少有一维相同的其他格子上的数都要大的话，我们就称它是极大的。

现在将 $1 \sim n * m * l$ 这 $n * m * l$ 个数等概率随机填入 $n * m * l$ 个格子（即任意数字出现在任意格子上的概率均相等），使得每个数恰出现一次，求恰有 k 个极大的数的概率。答案对 998244353（一个质数）取模。

【输入格式】

从文件 *cube.in* 中读入数据。

输入包含多组数据。输入第一行包含一个正整数 T ，表示数据组数。

接下来 T 行，每行一组数据，包含 4 个正整数 n, m, l, k ，表示一次询问。

【输出格式】

输出到文件 *cube.out* 中。

对于每次询问，输出一行一个整数，表示答案对 998244353 取模的余数。

可以证明，答案一定为有理数。设其为 a/b （ a 和 b 为互质的正整数，数据保证 b 不为 998244353 的倍数），则你需要保证输出的数 x 满足 $0 \leq x < 998244353$ 且 $a \equiv bx \pmod{998244353}$ 。可以证明这样的 x 唯一存在。

【样例 1 输入】

```
5
1 1 1 1
2 2 2 1
7 8 9 3
123 456 789 1
1000 1000 1000 10
```

【样例 1 输出】

```
1
142606337
736950806
246172965
189652652
```

【样例 2】

见选手目录下的 *cube/cube2.in* 与 *cube/cube2.ans*。

【测试数据约定】

对于 10% 的数据, $n, m \leq 2, l \leq 3, k = 1$ 。

对于 30% 的数据, $n, m, l, k \leq 12$ 。

对于 40% 的数据, $n, m, l \leq 100$ 。

对于 50% 的数据, $n, m, l \leq 1000$ 。

对于 60% 的数据, $n, m, l \leq 100000$, 其中有占全部数据 30% 的数据保证 $k = 1$ 。

对于 80% 的数据, $n, m, l \leq 1000000$, 其中有占全部数据 40% 的数据保证 $k = 1$ 。

对于 100% 的数据, $1 \leq n, m, l \leq 5000000, 1 \leq k \leq 100, 1 \leq T \leq 10$,

其中有 50% 的数据保证 $k = 1$ 。

珍珠 (pearl)

【题目背景】

白云苍狗，沧海桑田。

白云的眼前只剩下了模糊的一片。

在若隐若现之中，它看到了一个个小小的珍珠，有一些发着五彩的光芒。这些珍珠是白兔留下来的，每颗珍珠有一个颜色，为 D 种颜色中随机的一种。

白云想把这些珍珠放进一些小瓶子中，每个瓶子能恰好容纳两颗珍珠。不过它也有要求，每个瓶子必须装满，并且装的都是相同颜色的珍珠。

白云希望能得到至少 m 个装满珍珠的瓶子，它想知道它的愿望能被实现的概率是多少呢？

【题目描述】

有 n 个在范围 $[1, D]$ 内的整数均匀随机变量。

求至少能选出 m 个瓶子，使得存在一种方案，选择一些变量，并把选出来的每一个变量放到一个瓶子中，满足每个瓶子都恰好装两个值相同的变量的概率。

请输出概率乘上 D^n 后对 998244353 取模的值。取模部分说明可参考第一题。

【输入格式】

从文件 *pearl.in* 中读入数据。

输入仅一行，三个用空格隔开的整数 D, n, m

【输出格式】

输出到文件 *pearl.out* 中。

输出一个整数，表示所求概率乘上 D^n 后对 998244353 取模的结果。

【样例 1 输入】

2 2 1

【样例 1 输出】

2

【样例 1 解释】

情况 1：第一个变量为 1，第二个变量为 1

情况 2：第一个变量为 1，第二个变量为 2

情况 3：第一个变量为 2，第二个变量为 1

情况 4：第一个变量为 2，第二个变量为 2

其中情况 1 和 4 可以把两个变量放到一个瓶子中

情况 2 和 3 中两个变量的值不相同，所以不能放到同一个瓶子中

【样例 2】

见选手目录下的 *pearl/pearl2.in* 与 *pearl/pearl2.ans*。

【样例 3】

见选手目录下的 *pearl/pearl3.in* 与 *pearl/pearl3.ans*。

【测试数据约定】

测试点编号	D	n	m	
1	≤ 2	≤ 10	$\leq n$	
2		≤ 20		
3	≤ 100	≤ 100		
4				
5				
6				
7				
8	≤ 4000	≤ 4000		
9				
10				
11				
12				
13	≤ 300	≤ 10000000000	≤ 0	
14				
15				
16	≤ 100000		≤ 10000000000	≤ 1
17				≤ 2
18				$\leq n$
19				
20				
21				
22				
23				
24				
25				

所有测试点均满足 $1 \leq m \leq n \leq 10^9$ $1 \leq D \leq 10^5$

无处安放 (nowhere)

这是一道提交答案题。

【题目背景】

寂寥深夜，朦朦胧胧的月光为一切都笼上一层轻纱，你漫步在幽径，时而低头沉吟，时而凝望星空，只孤独与彷徨作伴，哪怕只言片语却也无人倾诉，万千思绪伴随着你炽热的心，随风而动，无处安放……

【题目描述】

你整理出了 n 条思绪，它们在你的心中是一个个矩形 r_i ，你想要在心中用一个大矩形 R 安放它们，也就是将 r_i 放置在 R 内部。为了保护这些思绪，它们安放的位置不能与其他思绪有重合部分，并且四条边要平行或垂直于 R 的边。两个思绪有重合部分，指它们安放位置的重合面积大于零。

你有两种安放方式：

1. 将 n 条思绪全部安放进 R 中，希望令 R 的面积尽量小。
2. 固定 R 的大小，希望将尽量多的思绪安放进 R 中。

现在我已知晓你整理好的思绪，为你选择了安放的方式，我想知道你心中最好的安放方案，请你告诉我。

【输入格式】

输入文件 `nowhere1.in ~ nowhere10.in` 已在试题目录下。

对于每组输入数据：

第一行两个整数 $type, n$ ，分别表示安放方式与思绪的个数。

若 $type = 2$ ，第二行两个整数 W, H ，表示 R 的边长。

接下来 n 行每行两个整数 w_i, h_i ，表示第 i 个思绪即 r_i 的边长。

同一行中输入的整数均以空格分隔。

【输出格式】

将输入文件对应的答案，输出到 `nowhere1.out ~ nowhere10.out` 中。

为了方便，若 R 的边长为 W, H ，我们将安放方案视作一个 $(0, 0) \sim (W, H)$ 的直角坐标系。注意对于 $type = 2$ 的测试点，你应按输入，将其视作 $(0, 0) \sim (W, H)$ 的坐标系，而不是 $(0, 0) \sim (H, W)$ 的坐标系。

输出共有 n 行，每行一或四个整数，描述第 i 个思绪即 r_i 的放置方案。

每行第一个整数 c_i ，其中 $c_i = 1$ 表示 r_i 被安放在 R 中； $c_i = 0$ 表示 r_i 未被安放在 R 中。

若 $c_i = 1$, 则该行接下来应输出三个整数 x_i, y_i, dir_i 。若 $dir_i = 0$, 则 r_i 放置在 $(x_i, y_i) \sim (x_i + w_i, y_i + h_i)$ 的矩形范围内; 若 $dir_i = 1$, 则 r_i 放置在 $(x_i, y_i) \sim (x_i + h_i, y_i + w_i)$ 的矩形范围内。

请注意确保你的输出格式正确, 且 $c_i, dir_i \in \{0, 1\}$ 。对于 $type = 1$ 的测试点, 所有 c_i 均应为 1。

同一行中输出的整数应以一个空格分隔开。

【样例 1 输入】

```
1 3
1 1
1 1
2 1
```

【样例 1 输出】

```
1 0 0 0
1 0 1 0
1 1 0 1
```

【样例 2 输入】

```
2 4
2 2
1 1
1 1
2 1
2 1
```

【样例 2 输出】

```
1 0 0 0
1 0 1 0
1 1 0 1
0
```

【评分方式】

评分准则 *nowhere1.ans* ~ *nowhere10.ans* 已放在试题目录下。

每个测试点设置了 10 个评分参数 a_1, a_2, \dots, a_{10} 。若选手输出不合法，则得零分。否则，对于安放方式 1，令 val 为 R 的面积，若 $val \leq a_i$ ，则你可获得 i 分；对于安放方式 2，令 val 为安放到 R 中的思绪个数，若 $val \geq a_i$ 则你可获得 i 分。满足多个得分条件，测试点得分取最高者。

【提示】

试题目录下提供有 checker 工具，用于检测你的输出文件。

使用它的方法是，在终端中运行

```
./checker <case_number>
```

其中 <case_number> 是测试数据的编号，例如

```
./checker 7
```

将测试 *nowhere7.out* 是否可以接受。

请保存备份好下发的 in 与 ans 文件，不要随意更改，防止 checker 出现不可预知的错误。