

- [NOI2013] 小Q的修炼

- 题目描述
- 输入格式
- 输出格式
- 样例 #1
 - 样例输入 #1
 - 样例输出 #1
- 提示
 - 评分标准
 - 如何测试你的输出
 - 更多功能
 - 特别提示

- 集合求和

- 题目描述
- 输入格式
- 输出格式
- 样例 #1
 - 样例输入 #1
 - 样例输出 #1
- 提示

- [NOI2016] 网格

- 题目描述
- 输入格式
- 输出格式
- 样例 #1
 - 样例输入 #1
 - 样例输出 #1
- 提示
 - 样例解释
 - 数据范围

- 地毯填补问题

- 题目描述
- 输入格式
- 输出格式
- 样例 #1
 - 样例输入 #1
 - 样例输出 #1

- 提示
- 【模板】快速排序
 - 题目描述
 - 输入格式
 - 输出格式
 - 样例 #1
 - 样例输入 #1
 - 样例输出 #1
 - 提示
- 黑匣子
 - 题目描述
 - 输入格式
 - 输出格式
 - 样例 #1
 - 样例输入 #1
 - 样例输出 #1
 - 提示
 - 数据规模与约定
- [NOIP2004 提高组] 合并果子 / [USACO06NOV] Fence Repair G
 - 题目描述
 - 输入格式
 - 输出格式
 - 样例 #1
 - 样例输入 #1
 - 样例输出 #1
 - 提示
- [JSOI2007] 建筑抢修
 - 题目描述
 - 输入格式
 - 输出格式
 - 样例 #1
 - 样例输入 #1
 - 样例输出 #1
 - 提示
- 舞蹈课
 - 题目描述
 - 输入格式
 - 输出格式

- 样例 #1
 - 样例输入 #1
 - 样例输出 #1
- 提示
- 无线通讯网
 - 题目描述
 - 输入格式
 - 输出格式
 - 样例 #1
 - 样例输入 #1
 - 样例输出 #1
 - 提示
- 买礼物
 - 题目描述
 - 输入格式
 - 输出格式
 - 样例 #1
 - 样例输入 #1
 - 样例输出 #1
 - 样例 #2
 - 样例输入 #2
 - 样例输出 #2
 - 提示
- 口袋的天空
 - 题目背景
 - 题目描述
 - 输入格式
 - 输出格式
 - 样例 #1
 - 样例输入 #1
 - 样例输出 #1
 - 提示
- [BJWC2010] 严格次小生成树
 - 题目描述
 - 输入格式
 - 输出格式
 - 样例 #1
 - 样例输入 #1

- 样例输出 #1
- 提示
- [JSOI2010]部落划分
 - 题目描述
 - 输入格式
 - 输出格式
 - 样例 #1
 - 样例输入 #1
 - 样例输出 #1
 - 样例 #2
 - 样例输入 #2
 - 样例输出 #2
 - 提示
 - 数据规模与约定
- 【模板】最小生成树
 - 题目描述
 - 输入格式
 - 输出格式
 - 样例 #1
 - 样例输入 #1
 - 样例输出 #1
 - 提示
- 最短路计数
 - 题目描述
 - 输入格式
 - 输出格式
 - 样例 #1
 - 样例输入 #1
 - 样例输出 #1
 - 提示

[NOI2013] 小Q的修炼

题目描述

小 Q 最近发现了一款新游戏，游戏的目标是从一个新手修炼成为武功高强的大侠。面对错综复杂的游戏世界，小 Q 要对他面临的每件事情做出谨慎的选择。例如，是否参加一

个陌生人邀请的比武；同意或是拒绝用宝剑交换他人的武功秘籍.....而小 Q 做出的每一个选择都有可能影响到他以后的发展：面对一个高手，若主动与之比武，很可能会损失惨重；但若不去比武，也许今后就再也见不到这个高手了。

对着这个游戏，小 Q 玩了很多次仍然玩不出他想要的结局，于是他费尽千辛万苦找到了游戏的剧本。令人惊讶的是,游戏的剧本并不像我们平时见到的剧本，反而很像代码。这个剧本是这样描述的：

- 量：有 2 种量，常数和变量。
- 常数：一个整数。
- 变量：初始值为 0 的可变整数，不同变量用不同正整数编号区分。
- 事件：整个剧本由若干个事件构成。所有的事件按照给定的顺序从 1 开始依次编号。事件共有 3 种：普通事件、选择跳转和条件跳转。
- 执行位置：一个整数，表示接下来将会执行的事件编号，如果不存在这个编号的事件则停止，即游戏到了一个结局。最初的时候执行位置为 1。
- 普通事件：一个变量增加或减少一个量的值。之后执行位置增加 1。
- 选择跳转：两个整数。执行到这里时玩家需要在这两个整数中选择一个，之后执行位置将被修改为这个整数。
- 条件跳转：两个量和两个整数。执行到这里时，若第一个量小于第二个量，则执行位置将被修改为第一个整数，否则将被修改为第二个整数。

小 Q 认为，整个游戏是希望一个叫做「成就值」的变量（编号为 1）最大。

输入格式

该题为提交答案型试题，所有输入数据 `train1.in~train10.in` 已在附加文件中。

输入的第一行包含两个正整数 n, m ，表示事件的个数和变量的个数。

接下来有 n 行，每行描述一个事件。这些事件按照给出的顺序依次编号为 1 到 n 。

描述量和事件的格式如下（格式中 #表示空格）

类型	格式	例子
常数	<code>c#整数</code>	<code>c -2</code>
变量	<code>v#正整数</code>	<code>v 5</code>
普通事件	<code>变量#+#量</code>	<code>v 1 + c 1</code>
普通事件	<code>变量#-#量</code>	<code>v 2 - c 2</code>

类型	格式	例子
选择跳转	s#整数 1#整数 2	s 10 20
条件跳转	i#量 1#量 2#整数 1#整数 2 i c 99 v 2 0 1	

输出格式

针对给定的 10 个输入文件 `train1.in~train10.in`，你需要分别提交你的输出文件 `train1.out~train10.out`。

每个文件需要输出若干行，每行输出一个字符 `1` 或 `2`，表示执行过程中遇到的每个选择跳转所作的选择。输出的行数需要严格等于此次游戏执行过程中遇到的选择跳转的个数。

样例 #1

样例输入 #1

```
11 2
v 2 + c 19
i v 2 c 3 7 3
s 4 7
v 1 + c 13
v 2 - c 3
i c 0 c 1 2 0
i v 2 c 5 12 8
s 9 12
v 1 + c 23
v 2 - c 5
i c 0 c 1 7 0
```

样例输出 #1

```
1
1
1
2
1
1
```

提示

评分标准

对于每组数据，我们采用如下方式评分：

- 如果你的输出不合法,得 0 分。
- 如果你的输出执行了超过 10^6 行剧本,得 0 分。
- 如果你的输出能让剧本正常结束，得 1 分。
- 如果你的输出能让剧本正常结束，且结束时成就值为正数，得 2 分。

我们设置了 8 个评分参数 a_3, a_4, \dots, a_{10} 。

如果你的输出能让剧本正常结束，且结束时成就值不小于 a_s ，得 s 分。

如果以上条目有多项满足，则取满足条件中的最高得分。

如何测试你的输出

我们提供 **checker** 这个工具来测试你的输出文件是否是可接受的。使用这个工具的方法是，首先进入终端，在终端中运行下面的命令进入本题的文件夹：

```
cd train
```

然后运行：

```
./checker <case_no>
```

其中 **case_no** 是测试数据的编号。例如

```
./checker 3
```

将测试 **train3.out** 是否可以接受。

在你调用这个程序后，**checker** 将根据你给出的输出文件给出测试的结果，其中包括：

- 非法退出：未知错误。
- **Input/Output file does not exist.**：输入/输出文件不存在。
- **Output invalid.**：输出文件有误,此时可能包含具体错误信息。
- **Correct! Your answer is x.**：输出可接受,最后的成就值为 x 。

更多功能

`checker` 还可以检查任意输入输出文件的测试结果，方法是在终端中运行：

```
cd train
```

```
./checker <input_file_name> <output_file_name>
```

其中 `input_file_name` 和 `output_file_name` 分别是输入输出文件的名称。例如

```
./checker train3.in train3.out
```

将测试 `train3.out` 是否可以接受。

使用 `-w` 可以输出每步运行的结果。用法是

```
./checker -w <input_file_name> <output_file_name>
```

或者

```
./checker -w <case_no>
```

例如

```
./checker -w train3.in train3.out
```

特别提示

如果选手使用自己生成输入文件进行调试，有可能因规模过大造成 `checker` 出错。若发生这类情况，请尝试较小规模的数据。

集合求和

题目描述

给定一个集合 S （集合元素数量 ≤ 30 ），求出此集合所有子集元素之和。

输入格式

集合中的元素（元素 ≤ 1000 ）

输出格式

s 所有子集元素之和。

样例 #1

样例输入 #1

2 3

样例输出 #1

10

提示

【样例解释】

子集为： $\emptyset, \{2\}, \{3\}, \{2, 3\}$ ，和为 $2 + 3 + 2 + 3 = 10$ 。

【数据范围】

对于 100% 的数据， $1 \leq |s| \leq 30$ ， $1 \leq s_i \leq 1000$ ， s 所有子集元素之和 $\leq 10^{18}$ 。

[NOI2016] 网格

题目描述

跳蚤国王和蛐蛐国王在玩一个游戏。

他们在一个 n 行 m 列的网格上排兵布阵。其中的 c 个格子中 ($0 \leq c \leq n \cdot m$)，每个格子有一只蛐蛐，其余的格子中，每个格子有一只跳蚤。

我们称占据的格子有公共边的两只跳蚤是相邻的。

我们称两只跳蚤是连通的，当且仅当这两只跳蚤相邻，或存在另一只跳蚤与这两只跳蚤都连通。

现在，蛐蛐国王希望，将某些（零个，一个或多个）跳蚤替换成蛐蛐，使得在此之后存在至少两只跳蚤不连通。

例如：图 1 描述了一个 $n = 4, m = 4, c = 2$ 的情况。

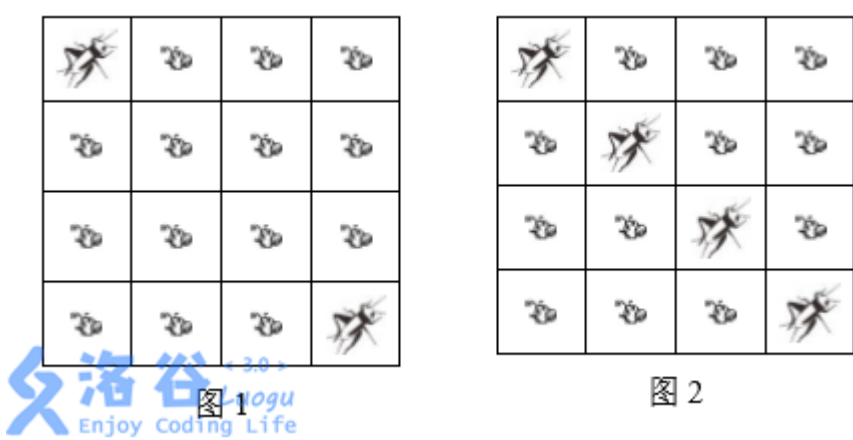


图 2

这种情况下蛐蛐国王可以通过将第二行第二列，和第三行第三列的两只跳蚤替换为蛐蛐，从而达成他的希望，如右图所示。并且，不存在更优的方案，但是可能存在其他替换两只跳蚤的方案。

你需要首先判断蛐蛐国王的希望能否被达成。如果能够达成，你还需要最小化被替换的跳蚤的个数。

输入格式

每个输入文件包含多组数据。

输入文件的第一行只有一个整数 T ，表示数据的组数。

接下来依次输入 T 组数据，每组数据的第一行包含三个整数 n, m, c 。

接下来 c 行，每行包含两个整数 x, y 表示第 x 行，第 y 列的格子被一个蛐蛐占据。每一组数据当中，同一个蛐蛐不会被多次描述。

输出格式

对于每一组数据依次输出一行答案。

如果这组数据中，蚰蚰国王的希望不能被达成，输出 -1 。否则，输出被替换的跳蚤的个数的最小值。

样例 #1

样例输入 #1

```
4
4 4 2
1 1
4 4
2 3 1
1 2
2 2 2
1 1
2 2
1 1 0
```

样例输出 #1

```
2
1
0
-1
```

提示

样例解释

第一组数据就是问题描述中的例子。

对于第二组数据，可以将第二行第二列的一只跳蚤替换为蛐蛐，从而使得存在两只跳蚤不连通，并且不存在更优的方案。

对于第三组数据，最初已经存在两只跳蚤不连通，故不需要再进行替换。

对于第四组数据，由于最多只有一只跳蚤，所以无论如何替换都不能存在两只跳蚤不连通。

数据范围

对于全部的测试点，保证 $1 \leq T \leq 20$ 。我们记 $\sum c$ 为某个测试点中，其 T 组输入数据的所有 c 的总和。对于所有的测试点， $\sum c \leq 10^5$ 。

对于全部的数据，满足 $1 \leq n, m \leq 10^9$ ， $0 \leq c \leq n \times m$ ， $1 \leq x \leq n, 1 \leq y \leq m$ 。

每个测试点的详细数据范围见下表。表中的 n, m, c 均是对单个输入数据（而非测试点）而言的，也就是说同一个测试点下的 T 组数据均满足限制条件；而 $\sum c$ 是对单个测试点而言的。为了方便阅读，“测试点”一列被放到了表格的中间而不是左边。

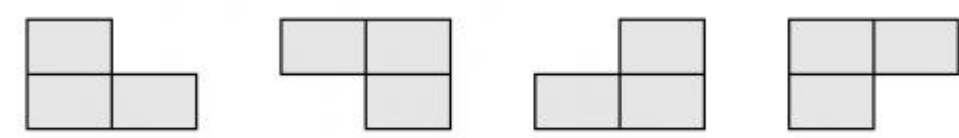
n, m	测试点	c
$n * m \leq 4$	1	$c \leq n * m$
$n * m \leq 8$	2	$c \leq n * m$
$n * m \leq 15$	3	$c \leq n * m$
$n * m \leq 30$	4	$c \leq n * m$
$n * m \leq 100$	5	$c \leq n * m$
$n * m \leq 300$	6	$c \leq n * m$
$n * m \leq 10^3$	7	$c \leq n * m$
$n * m \leq 2 \times 10^4$	8	$c \leq 5$
$n * m \leq 2 \times 10^4$	9	$c \leq 15$
$n * m \leq 2 \times 10^4$	10	$c \leq 30$
$n, m \leq 2 \times 10^4, n * m \leq 2 \times 10^4$	11	$\sum c \leq 2 \times 10^4$
$n, m \leq 2 \times 10^4, n * m \leq 10^5$	12	$\sum c \leq 2 \times 10^4$
$n, m \leq 2 \times 10^4, n * m \leq 3 \times 10^5$	13	$\sum c \leq 2 \times 10^4$
$n, m \leq 2 \times 10^4, n * m \leq 10^6$	14	$\sum c \leq 2 \times 10^4$

n, m	测试点	c
$n, m \leq 2 \times 10^4, n * m \leq 10^9$	15	$\sum c \leq 2 \times 10^4$
$n, m \leq 10^5$	16	$\sum c \leq 10^5$
$n, m \leq 10^9$	17	$c = 0$
$n, m \leq 10^9$	18	$c \leq 1$
$n, m \leq 10^9$	19	$c \leq 2$
$n, m \leq 10^9$	20	$c \leq 3$
$n, m \leq 10^9$	21	$c \leq 10$
$n, m \leq 10^9$	22	$c \leq 30$
$n, m \leq 10^9$	23	$c \leq 300$
$n, m \leq 10^9$	24	$\sum c \leq 2 \times 10^4$
$n, m \leq 10^9$	25	$\sum c \leq 10^5$

地毯填补问题

题目描述

相传在一个古老的阿拉伯国家里，有一座宫殿。宫殿里有个四四方方的格子迷宫，国王选择驸马的方法非常特殊，也非常简单：公主就站在其中一个方格子上，只要谁能用地毯将除公主站立的地方外的所有地方盖上，美丽漂亮聪慧的公主就是他的人了。公主这一个方格不能用地毯盖住，毯子的形状有所规定，只能有四种选择（如图）：



并且每一方格只能用一层地毯，迷宫的大小为 $2^k \times 2^k$ 的方形。当然，也不能让公主无限限制的在那儿等，对吧？由于你使用的是计算机，所以实现时间为 1s。

输入格式

输入文件共 2 行。

第一行： k ，即给定被填补迷宫的大小为 $2^k \times 2^k$ ($0 < k \leq 10$)；

第二行： x,y ，即给出公主所在方格的坐标（ x 为行坐标， y 为列坐标）， x 和 y 之间有一个空格隔开。

输出格式

将迷宫填补完整的方案：每一补(行)为 $x\ y\ c$ （ x,y 为毯子拐角的行坐标和列坐标， c 为使用毯子的形状，具体见上面的图 1，毯子形状分别用 1,2,3,4 表示， x,y,c 之间用一个空格隔开）。

样例 #1

样例输入 #1

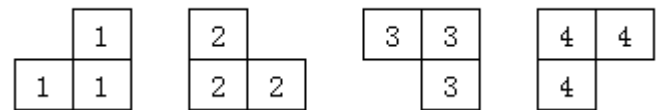
```
3
3 3
```

样例输出 #1

```
5 5 1
2 2 4
1 1 4
1 4 3
4 1 2
4 4 1
2 7 3
1 5 4
1 8 3
3 6 3
4 8 1
7 2 2
5 1 4
6 3 2
8 1 2
8 4 1
7 7 1
6 6 1
5 8 3
8 5 2
8 8 1
```

提示

事实上感觉四个的形状分别是这样（仅供参考，如果有问题联系 icy）



spj 报错:

1. c 越界
2. x, y 越界
3. $mp[x][y]$ 已被占用
4. $mp[x][y]$ 从未被使用

【模板】快速排序

题目描述

利用快速排序算法将读入的 N 个数从小到大排序后输出。

快速排序是信息学竞赛的必备算法之一。对于快速排序不是很了解的同学可以自行上网查询相关资料，掌握后独立完成。（C++ 选手请不要试图使用 STL，虽然你可以使用 `sort` 一遍过，但是你并没有掌握快速排序算法的精髓。）

输入格式

第 1 行为一个正整数 N ，第 2 行包含 N 个空格隔开的正整数 a_i ，为你需要进行排序的数，数据保证了 A_i 不超过 10^9 。

输出格式

将给定的 N 个数从小到大输出，数之间空格隔开，行末换行且无空格。

样例 #1

样例输入 #1

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

样例输出 #1

```
1 2 4 4 5
```

提示

对于 20% 的数据，有 $N \leq 10^3$ ；

对于 100% 的数据，有 $N \leq 10^5$ 。

黑匣子

题目描述

Black Box 是一种原始的数据库。它可以储存一个整数数组，还有一个特别的变量 i 。最开始的时候 **Black Box** 是空的，而 $i = 0$ 。这个 **Black Box** 要处理一串命令。

命令只有两种：

- **ADD(x)**：把 x 元素放进 **Black Box**；
- **GET**： i 加 1，然后输出 **Black Box** 中第 i 小的数。

记住：第 i 小的数，就是 **Black Box** 里的数的按从小到大的顺序排序后的第 i 个元素。

我们来演示一下一个有 11 个命令的命令串。（如下表所示）

序号	操作	i	数据库	输出
1	ADD(3)	0	3	/
2	GET	1	3	3
3	ADD(1)	1	1, 3	/
4	GET	2	1, 3	3
5	ADD(-4)	2	-4, 1, 3	/
6	ADD(2)	2	-4, 1, 2, 3	/
7	ADD(8)	2	-4, 1, 2, 3, 8	/
8	ADD(-1000)	2	-1000, -4, 1, 2, 3, 8	/
9	GET	3	-1000, -4, 1, 2, 3, 8	1
10	GET	4	-1000, -4, 1, 2, 3, 8	2
11	ADD(2)	4	-1000, -4, 1, 2, 2, 3, 8	/

现在要求找出对于给定的命令串的最好的处理方法。ADD 命令共有 m 个，GET 命令共有 n 个。现在用两个整数数组来表示命令串：

- a_1, a_2, \cdots, a_m ：一串将要被放进 Black Box 的元素。例如上面的例子中 $a = [3, 1, -4, 2, 8, -1000, 2]$ 。
- u_1, u_2, \cdots, u_n ：表示第 u_i 个元素被放进了 Black Box 里后就出现一个 GET 命令。例如上面的例子中 $u = [1, 2, 6, 6]$ 。输入数据不用判错。

输入格式

第一行两个整数 m 和 n ，表示元素的个数和 GET 命令的个数。

第二行共 m 个整数，从左至右第 i 个整数为 a_i ，用空格隔开。

第三行共 n 个整数，从左至右第 i 个整数为 u_i ，用空格隔开。

输出格式

输出 Black Box 根据命令串所得出的输出串，一个数字一行。

样例 #1

样例输入 #1

```
7 4
3 1 -4 2 8 -1000 2
1 2 6 6
```

样例输出 #1

```
3
3
1
2
```

提示

数据规模与约定

- 对于 30% 的数据， $1 \leq n, m \leq 10^4$ 。
- 对于 50% 的数据， $1 \leq n, m \leq 10^5$ 。
- 对于 100% 的数据， $1 \leq n, m \leq 2 \times 10^5$, $|a_i| \leq 2 \times 10^9$ ，保证 u 序列单调不降。

[NOIP2004 提高组] 合并果子 / [USACO06NOV] Fence Repair G

题目描述

在一个果园里，多多已经将所有的果子打了下来，而且按果子的不同种类分成了不同的堆。多多决定把所有的果子合成一堆。

每一次合并，多多可以把两堆果子合并到一起，消耗的体力等于两堆果子的重量之和。可以看出，所有的果子经过 $n - 1$ 次合并之后，就只剩下一堆了。多多在合并果子时总共消耗的体力等于每次合并所耗体力之和。

因为还要花大力气把这些果子搬回家，所以多多在合并果子时要尽可能地节省体力。假定每个果子重量都为 1，并且已知果子的种类数和每种果子的数目，你的任务是设计出合并的次序方案，使多多耗费的体力最少，并输出这个最小的体力耗费值。

例如有 3 种果子，数目依次为 1，2，9。可以先将 1、2 堆合并，新堆数目为 3，耗费体力为 3。接着，将新堆与原先的第三堆合并，又得到新的堆，数目为 12，耗费体力为 12。所以多多总共耗费体力 $= 3 + 12 = 15$ 。可以证明 15 为最小的体力耗费值。

输入格式

共两行。第一行是一个整数 $n(1 \leq n \leq 10000)$ ，表示果子的种类数。

第二行包含 n 个整数，用空格分隔，第 i 个整数 $a_i(1 \leq a_i \leq 20000)$ 是第 i 种果子的数目。

输出格式

一个整数，也就是最小的体力耗费值。输入数据保证这个值小于 2^{31} 。

样例 #1

样例输入 #1

```
3
1 2 9
```

样例输出 #1

```
15
```

提示

对于 30% 的数据，保证有 $n \leq 1000$ ；

对于 50% 的数据，保证有 $n \leq 5000$ ；

对于全部的数据，保证有 $n \leq 10000$ 。

[JSOI2007] 建筑抢修

题目描述

小刚在玩 JSOI 提供的一个称之为“建筑抢修”的电脑游戏：经过了一场激烈的战斗，T 部落消灭了所有 Z 部落的入侵者。但是 T 部落的基地里已经有 N 个建筑设施受到了严重的损伤，如果不尽快修复的话，这些建筑设施将会完全毁坏。现在的情况是：T 部落基地里只有一个修理工人，虽然他能瞬间到达任何一个建筑，但是修复每个建筑都需要一定的时间。同时，修理工人修理完一个建筑才能修理下一个建筑，不能同时修理多个建筑。如果某个建筑在一段时间之内没有完全修理完毕，这个建筑就报废了。你的任务是帮小刚合理的制订一个修理顺序，以抢修尽可能多的建筑。

输入格式

第一行，一个整数 N 。

接下来 N 行，每行两个整数 T_1, T_2 描述一个建筑：修理这个建筑需要 T_1 秒，如果在 T_2 秒之内还没有修理完成，这个建筑就报废了。

输出格式

输出一个整数 S ，表示最多可以抢修 S 个建筑。

样例 #1

样例输入 #1

```
4
100 200
200 1300
1000 1250
2000 3200
```

样例输出 #1

```
3
```

提示

对于 100% 的数据， $1 \leq N < 150000$ ， $1 \leq T_1 < T_2 < 2^{31}$ 。

舞蹈课

题目描述

有 n 个人参加一个舞蹈课。每个人的舞蹈技术由整数来决定。在舞蹈课的开始，他们从左到右站成一排。当这一排中至少有一对相邻的异性时，舞蹈技术相差最小的那一对会出列并开始跳舞。如果不止一对，那么最左边的那一对出列。一对异性出列之后，队伍中的空白按原顺序补上（即：若队伍为 **ABCD**，那么 **BC** 出列之后队伍变为 **AD**）。舞蹈技术相差最小即是 a_i 的绝对值最小。

任务是模拟以上过程，确定跳舞的配对及顺序。

输入格式

- 第一行一个正整数 n 表示队伍中的人数。
- 第二行包含 n 个字符 **B** 或者 **G**，**B** 代表男，**G** 代表女。

第三行为 n 个整数 a_i 。所有信息按照从左到右的顺序给出。

输出格式

第一行一个整数表示出列的总对数 k 。

接下来 k 行，每行是两个整数。按跳舞顺序输出，两个整数代表这一对舞伴的编号（按输入顺序从左往右 1 至 n 编号）。请先输出较小的整数，再输出较大的整数。

样例 #1

样例输入 #1

```
4
BGBG
4 2 4 3
```

样例输出 #1

```
2
3 4
1 2
```

提示

对于 50% 的数据， $1 \leq n \leq 200$ 。

对于 100% 的数据， $1 \leq n \leq 2 \times 10^5$ ， $1 \leq a_i \leq 10^7$ 。

无线通讯网

题目描述

国防部计划用无线网络连接若干个边防哨所。2 种不同的通讯技术用来搭建无线网络；每个边防哨所都要配备无线电收发器；有一些哨所还可以增配卫星电话。

任意两个配备了一条卫星电话线路的哨所（两边都 有卫星电话）均可以通话，无论他们相距多远。而只通过无线电收发器通话的哨所之间的距离不能超过 D ，这是受收发器的功率限制。收发器的功率越高，通话距离 D 会更远，但同时价格也会更贵。

收发器需要统一购买和安装，所以全部哨所只能选择安装一种型号的收发器。换句话说，每一对哨所之间的通话距离都是同一个 D 。你的任务是确定收发器必须的最小通话距离 D ，使得每一对哨所之间至少有一条通话路径（直接的或者间接的）。

输入格式

从 `wireless.in` 中输入数据第 1 行，2 个整数 S 和 P ， S 表示可安装的卫星电话的哨所数， P 表示边防哨所的数量。接下里 P 行，每行两个整数 x, y 描述一个哨所的平面坐标 (x,y) ，以 km 为单位。

输出格式

输出 `wireless.out` 中

第 1 行，1 个实数 D ，表示无线电收发器的最小传输距离，精确到小数点后两位。

样例 #1

样例输入 #1

```
2 4
0 100
0 300
0 600
150 750
```

样例输出 #1

提示

对于 20% 的数据： $P = 2$ ， $S = 1$

对于另外 20% 的数据： $P = 4$ ， $S = 2$

对于 100% 的数据保证： $1 \leq S \leq 100$ ， $S < P \leq 500$ ， $0 \leq x, y \leq 10000$ 。

买礼物

题目描述

又到了一年一度的明明生日了，明明想要买 B 样东西，巧的是，这 B 样东西价格都是 A 元。

但是，商店老板说最近有促销活动，也就是：

如果你买了第 I 样东西，再买第 J 样，那么就可以只花 $K_{I,J}$ 元，更巧的是， $K_{I,J}$ 竟然等于 $K_{J,I}$ 。

现在明明想知道，他最少要花多少钱。

输入格式

第一行两个整数， A, B 。

接下来 B 行，每行 B 个数，第 I 行第 J 个为 $K_{I,J}$ 。

我们保证 $K_{I,J} = K_{J,I}$ 并且 $K_{I,I} = 0$ 。

特别的，如果 $K_{I,J} = 0$ ，那么表示这两样东西之间不会导致优惠。

输出格式

一个整数，为最小要花的钱数。

样例 #1

样例输入 #1

```
1 1
0
```

样例输出 #1

```
1
```

样例 #2

样例输入 #2

```
3 3
0 2 4
2 0 2
4 2 0
```

样例输出 #2

```
7
```

提示

样例解释 2。

先买第 2 样东西，花费 3 元，接下来因为优惠，买 1,3 样都只要 2 元，共 7 元。

（同时满足多个“优惠”的时候，聪明的明明当然不会选择用 4 元买剩下那件，而选择用 2 元。）

数据规模

对于 30% 的数据， $1 \leq B \leq 10$ 。

对于 100% 的数据， $1 \leq B \leq 500, 0 \leq A, K_{I,J} \leq 1000$ 。

2018.7.25新添数据一组

口袋的天空

题目背景

小杉坐在教室里，透过口袋一样的窗户看口袋一样的天空。

有很多云飘在那里，看起来很漂亮，小杉想摘下那样美的几朵云，做成棉花糖。

题目描述

给你云朵的个数 N ，再给你 M 个关系，表示哪些云朵可以连在一起。

现在小杉要把所有云朵连成 K 个棉花糖，一个棉花糖最少要用掉一朵云，小杉想知道他怎么连，花费的代价最小。

输入格式

第一行有三个数 N, M, K 。

接下来 M 行每行三个数 X, Y, L ，表示 X 云和 Y 云可以通过 L 的代价连在一起。

输出格式

对每组数据输出一行，仅有一个整数，表示最小的代价。

如果怎么连都连不出 K 个棉花糖，请输出 **No Answer**。

样例 #1

样例输入 #1

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

样例输出 #1

```
1
```

提示

对于 30% 的数据， $1 \leq N \leq 100$ ， $1 \leq M \leq 10^3$ ；

对于 100% 的数据， $1 \leq N \leq 10^3$ ， $1 \leq M \leq 10^4$ ， $1 \leq K \leq 10$ ， $1 \leq X, Y \leq N$ ， $0 \leq L < 10^4$ 。

[BJWC2010] 严格次小生成树

题目描述

小 C 最近学了很多最小生成树的算法，Prim 算法、Kruskal 算法、消圈算法等等。正当小 C 洋洋得意之时，小 P 又来泼小 C 冷水了。小 P 说，让小 C 求出一个无向图的次小生成树，而且这个次小生成树还得是严格次小的，也就是说：如果最小生成树选择的边集是 E_M ，严格次小生成树选择的边集是 E_S ，那么需要满足： $(value(e)$ 表示边 e 的权值) $\sum_{e \in E_M} value(e) < \sum_{e \in E_S} value(e)$

这下小 C 蒙了，他找到了你，希望你帮他解决这个问题。

输入格式

第一行包含两个整数 N 和 M ，表示无向图的点数与边数。

接下来 M 行，每行 3 个数 x,y,z 表示，点 x 和点 y 之间有一条边，边的权值为 z 。

输出格式

包含一行，仅一个数，表示严格次小生成树的边权和。

样例 #1

样例输入 #1

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

样例输出 #1

```
11
```

提示

数据中无向图不保证无自环

对于 50% 的数据， $N \leq 2000$ ， $M \leq 3000$ 。

对于 80% 的数据， $N \leq 5 \times 10^4$ ， $M \leq 10^5$ 。

对于 100% 的数据， $N \leq 10^5$ ， $M \leq 3 \times 10^5$ ，边权 $\in [0, 10^9]$ ，数据保证必定存在严格次小生成树。

[JSOI2010]部落划分

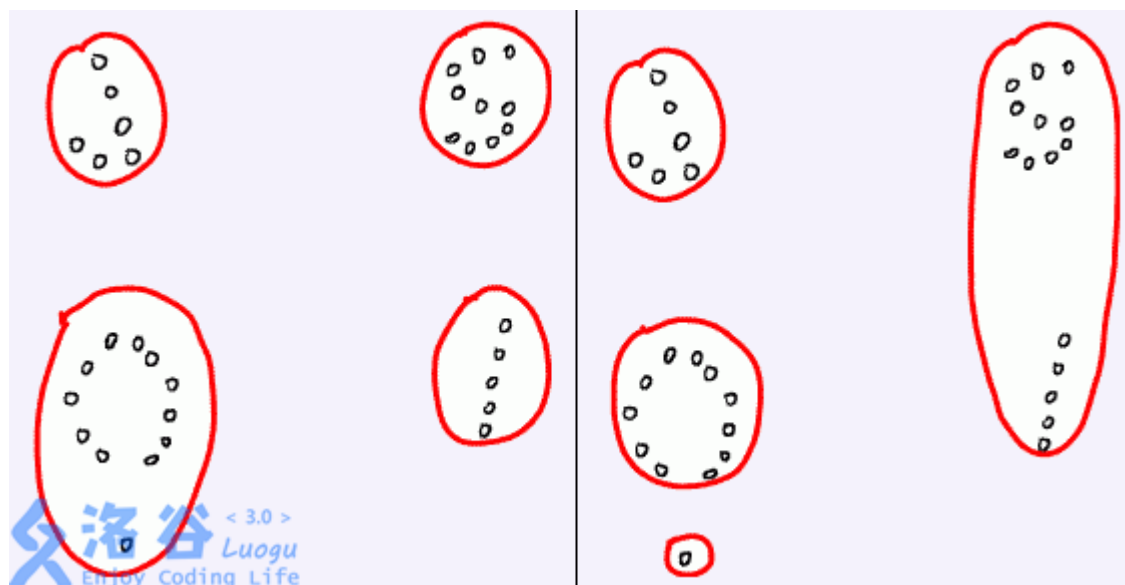
题目描述

聪聪研究发现，荒岛野人总是过着群居的生活，但是，并不是整个荒岛上的所有野人都属于同一个部落，野人们总是拉帮结派形成属于自己的部落，不同的部落之间则经常发生争斗。只是，这一切都成为谜团了——聪聪根本就不知道部落究竟是如何分布的。

不过好消息是，聪聪得到了一份荒岛的地图。地图上标注了 n 个野人居住的地点（可以看作是平面上的坐标）。我们知道，同一个部落的野人总是生活在附近。我们把两个部落的距离，定义为部落中距离最近的那两个居住点的距离。聪聪还获得了一个有意义的信息——这些野人总共被分为了 k 个部落！这真是个好消息。聪聪希望从这些信息里挖掘出所有部落的详细信息。他正在尝试这样一种算法：

对于任意一种部落划分的方法，都能够求出两个部落之间的距离，聪聪希望求出一种部落划分的方法，使靠得最近的两个部落尽可能远离。

例如，下面的左图表示了一个好的划分，而右图则不是。请你编程帮助聪聪解决这个难题。



输入格式

输入文件第一行包含两个整数 n 和 k ，分别代表了野人居住点的数量和部落的数量。

接下来 n 行，每行包含两个整数 x, y ，描述了一个居住点的坐标。

输出格式

输出一行一个实数，为最优划分时，最近的两个部落的距离，精确到小数点后两位。

样例 #1

样例输入 #1

```
4 2
0 0
0 1
1 1
1 0
```

样例输出 #1

```
1.00
```

样例 #2

样例输入 #2

```
9 3
2 2
2 3
3 2
3 3
3 5
3 6
4 6
6 2
6 3
```

样例输出 #2

2.00

提示

数据规模与约定

对于 100% 的数据，保证 $2 \leq k \leq n \leq 10^3$ ， $0 \leq x, y \leq 10^4$ 。

【模板】最小生成树

题目描述

如题，给出一个无向图，求出最小生成树，如果该图不连通，则输出 **orz**。

输入格式

第一行包含两个整数 N, M ，表示该图共有 N 个结点和 M 条无向边。

接下来 M 行每行包含三个整数 X_i, Y_i, Z_i ，表示有一条长度为 Z_i 的无向边连接结点 X_i, Y_i 。

输出格式

如果该图连通，则输出一个整数表示最小生成树的各边的长度之和。如果该图不连通则输出 **orz**。

样例 #1

样例输入 #1

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

样例输出 #1

7

提示

数据规模：

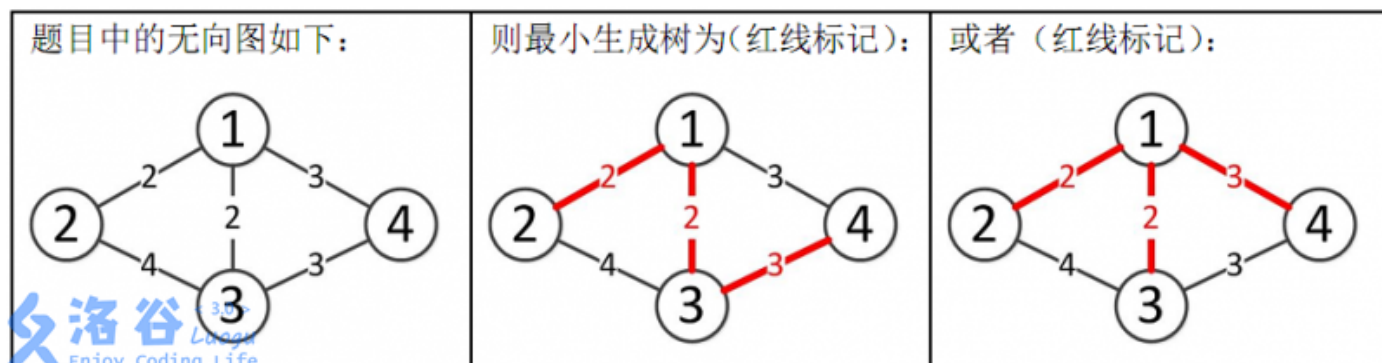
对于 20% 的数据， $N \leq 5$ ， $M \leq 20$ 。

对于 40% 的数据， $N \leq 50$ ， $M \leq 2500$ 。

对于 70% 的数据， $N \leq 500$ ， $M \leq 10^4$ 。

对于 100% 的数据： $1 \leq N \leq 5000$ ， $1 \leq M \leq 2 \times 10^5$ ， $1 \leq Z_i \leq 10^4$ 。

样例解释：



所以最小生成树的总边权为 $2 + 2 + 3 = 7$ 。

最短路径计数

题目描述

给出一个 N 个顶点 M 条边的无向无权图，顶点编号为 $1 \sim N$ 。问从顶点 1 开始，到其他每个点的最短路有几条。

输入格式

第一行包含 2 个正整数 N, M ，为图的顶点数与边数。

接下来 M 行，每行 2 个正整数 x, y ，表示有一条由顶点 x 连向顶点 y 的边，请注意可能有自环与重边。

输出格式

共 N 行，每行一个非负整数，第 i 行输出从顶点 1 到顶点 i 有多少条不同的最短路，由于答案有可能会很大，你只需要输出 $\text{\$ ans \bmod 100003\$}$ 后的结果即可。如果无法到达顶点 i 则输出 0。

样例 #1

样例输入 #1

```
5 7
1 2
1 3
2 4
3 4
2 3
4 5
4 5
```

样例输出 #1

1
1
1
2
4

提示

1 到 5 的最短路有 4 条，分别为 2 条 $1 \rightarrow 2 \rightarrow 4 \rightarrow 5$ 和 2 条 $1 \rightarrow 3 \rightarrow 4 \rightarrow 5$ （由于 $4 \rightarrow 5$ 的边有 2 条）。

对于 20% 的数据， $1 \leq N \leq 100$ ； 对于 60% 的数据， $1 \leq N \leq 10^3$ ； 对于 100% 的数据， $1 \leq N \leq 10^6$ ， $1 \leq M \leq 2 \times 10^6$ 。