

Sunny bunny 的 Fall Semester

Duration: 4h

题目概况

题目名称	军训	秋游	TPT	元旦晚会
英文名称	train	tour	tpt	party
源程序名	train.cpp	tour.cpp	tpt.cpp	party.cpp
可执行文件名	train	tour	tpt	party
输入文件名	train.in	tour.in	tpt.in	party.in
输出文件名	train.out	tour.out	tpt.out	party.out
时间限制	1 s	1 s	2 s	1 s
空间限制	512 MB	512 MB	512 MB	512 MB
测试点/组数目	25	25	3	50
是否有附加文件	否	是	是	是
结果比较方式	Special Judge	-	-	-

测试统一编译命令如下：

- C++ 选手：

```
g++ ${源程序名} -o ${可执行文件名} -lm -std=c++14 -O2
```

注意事项

- 选手提交的源文件必须存放在选手目录下，不建立子文件夹。
- 文件名（源文件名和输入输出文件名）必须使用英文小写。
- C/C++ 中 main() 函数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。
- 若无特殊说明，输入文件中同一行内的多个整数、浮点数、字符串等均使用一个空格进行分隔。
- 若无特殊说明，结果比较方式为忽略行末空格、文末回车后的全文比较。
- 评测在当前最新公布的 NOI Linux 下进行，各语言编译器版本以其为准。

军训

(train.cpp/.in/.out)

题目背景

一转眼就快开学了, *Sunny bunny* 要迎接的第一项挑战便是开学前的军训。

题目描述

军训的第一项是列队, 以及站军姿。

现在, 参加军训的有 n 只兔子, 每只兔子按照它的身高得到了一个与众不同的编号。从左往右数, 第 k 只兔子的编号为 h_k 。

教官需要将兔子们按照身高排好序。这一次他想尝试一种全新的排序方式:

- 每次重新列队之前, 教官会先选定一个整数 k , 满足 $2 \leq k \leq n$ 。
- 教官将队列从左到右划分为 k 段。从左到右第 i 段被标记为 s_i 。显然, 这样划分会使得 $\sum_{i=1}^k |s_i| = n$ 。
- 原先的队列为 $\{s_1, s_2, \dots, s_k\}$, 新的队列会变成 $\{s_k, s_{k-1}, s_{k-2}, \dots, s_2, s_1\}$, 也就是按段反序。
- 教官会持续重新列队直到队列排好序。

站在队列里的 *Sunny bunny* 想要知道, 在这样的排序方式之下, 如何才能将队列排好序, 即从左往右 h 为升序呢?

输入格式

第一行输入两个正整数 n, w 。 n 的含义详见题目描述, w 会用于评估你的得分。

第二行输入 n 个正整数, 第 i 个正整数为 h_i 。

输出格式

首先, 你需要输出一个非负整数 q , 表示你的方案所需的重新列队次数。

接下来应有 q 行, 其中第 i 行应首先包含一个大于 1 的整数 k , 表示第 i 次列队的划分段数; 该行的剩余部分应当包含 k 个正整数, 分别为 l_1, l_2, \dots, l_k , 表示每一段的长度, 且应当满足 $\sum_{i=1}^k l_i = n$ 。

你的输出应当只包含空格、数字和 `\t, \r, \n` 几种字符, 否则可能导致误判。

评分细则

如果你的输出不合法，则该测试点获得 0 分的好成绩！

如果你的输出合法，但是 $q > w$ ，则该测试点获得 40% 的分数。

如果你的输出合法，且 $q \leq w$ ，则该测试点获得 100% 的分数。

测试用例

本题中，测试样例仅供参考。

样例 1 输入

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

样例 1 输出

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

样例 1 解释

初始时，队列为 $\{3, 1, 2, 4\}$ 。

第一次操作后，队列为 $\{4, 1, 2, 3\}$ 。

第二次操作后，队列为 $\{1, 2, 3, 4\}$ 。

总共的重新列队次数为 2 次。

样例 2 输入

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

样例 2 输出

```
1
6 1 1 1 1 1 1
```

样例 3 输入

```
1 2
1
```

样例 3 输出

```
0
```

数据规模与约定

对于 100% 的数据，满足 $1 \leq n \leq 3000, n \leq w \leq n^2, 1 \leq h_k \leq n$ ，且 h_k 互不相同。

其余约定请见下表：

测试点编号	n	w
1 ~ 4	≤ 10	$= n^2$
5 ~ 8	≤ 300	$= 8n$
9 ~ 12	≤ 3000	$= 2n$
13 ~ 25	≤ 3000	$= n$

秋游

(tour.cpp/.in/.out)

题目背景

人出门旅行并不是为了到达某地，而是为了旅游。

——歌德

题目描述

Sunny bunny 迎来了学校的秋季出游活动，他决定乘坐火车旅行。

Sunny bunny 所在的区域有 n 个参观地点，有 m 条双向铁路连接着这 n 个参观地点。保证从任何一个参观地点出发，经由铁路可以到达其余任何一个参观地点。对参观地点和铁路编号后，第 i 条铁路连接了地点 u_i 和地点 v_i ，乘坐火车经过该铁路的花费为 w_i 。

铁路公司目前正在举行降价活动。对于一种乘坐火车的路线 $\{p_0, e_1, p_1, e_2, p_2, e_3, \dots, e_{k-1}, p_{k-1}, e_k, p_k\}$ ，铁路公司将票价定为：

$$\sum_{i=1}^k w_{e_i} - \max_{1 \leq i \leq k} w_{e_i} + \min_{1 \leq i \leq k} w_{e_i}$$

显然，现在的票价不会比降价之前的贵。

Sunny bunny 的学校在参观地点 1 的附近，因此他需要在那里开始乘坐火车。他希望知道，从地点 1 出发，只按照一条完整的路线乘火车（也即中途不下车），到达其余每个地点的最小票价是多少？

一条乘坐火车的路线被定义为一个地点编号和铁路编号的交错序列 $\{p_0, e_1, p_1, e_2, p_2, e_3, \dots, e_{k-1}, p_{k-1}, e_k, p_k\}$ 。其中 k 为某一个非负整数，对于任意的 $1 \leq j \leq k$ ， e_j 为某条边的编号；对于任意的 $0 \leq j \leq k$ ， p_j 为某个地点的编号。

此外，该序列还应该满足对于任意的 $1 \leq j \leq k$ ， $\{p_{j-1}, p_j\} = \{u_{e_j}, v_{e_j}\}$ 。

当 Sunny bunny 按这条路线乘坐火车的时候，他会从 p_0 出发，在 p_k 处下火车。

输入格式

第一行输入两个正整数 n, m 。

接下来的 m 行，第 i 行输入三个正整数 u_i, v_i, w_i 。

输出格式

输出一行 $n - 1$ 个非负整数，其中第 i 个非负整数表示从地点 1 出发到达地点 $i + 1$ 的最小票价。

测试用例

样例 1 输入

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

样例 1 输出

```
1 2 2 4
```

样例 2 输入

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

样例 2 输出

```
2 1 4 3 1
```

样例 3 输入

```
7 10
7 5 5
2 3 3
4 7 1
5 3 6
2 7 6
6 2 6
3 7 6
4 2 1
3 1 4
1 7 4
```

样例 3 输出

```
3 4 2 7 7 3
```

样例 3 解释

到达地点 2 的最优路线为 $\{1, 10, 7, 3, 4, 8, 2\}$ ，票价为 $4 + 1 + 1 - 4 + 1 = 3$ 。

到达地点 3 的最优路线为 $\{1, 9, 3\}$ ，票价为 $3 - 3 + 3 = 3$ 。

到达地点 4 的最优路线为 $\{1, 10, 7, 3, 4\}$ ，票价为 $4 + 1 - 4 + 1 = 2$ 。

到达地点 5 的最优路线为 $\{1, 10, 7, 3, 4, 3, 7, 1, 5\}$ ，票价为 $4 + 1 + 1 + 5 - 5 + 1 = 7$ 。

到达地点 6 的最优路线为 $\{1, 10, 7, 3, 4, 8, 2, 6, 6\}$ ，票价为 $4 + 1 + 1 + 6 - 6 + 1 = 7$ 。

到达地点 7 的最优路线为 $\{1, 10, 7, 3, 4, 3, 7\}$ ，票价为 $4 + 1 + 1 - 4 + 1 = 3$ 。

样例 4

详见下发文件中的 `tour/tour4.in` 和 `tour/tour4.out`。

该样例满足测试点 5 的数据约定。

样例 5

详见下发文件中的 `tour/tour5.in` 和 `tour/tour5.out`。

该样例满足测试点 11 的数据约定。

样例 6

详见下发文件中的 `tour/tour6.in` 和 `tour/tour6.out`。

该样例满足测试点 15 的数据约定。

数据规模与约定

对于 100% 的数据，满足 $1 \leq n, m \leq 2 \times 10^5, 1 \leq u_i, v_i \leq n, u_i \neq v_i, 1 \leq w_i \leq 10^9$ ，且对于任意的 $1 \leq i < j \leq m$ ，都有 $\{u_i, v_i\} \neq \{u_j, v_j\}$ 。

其余约定请见下表：

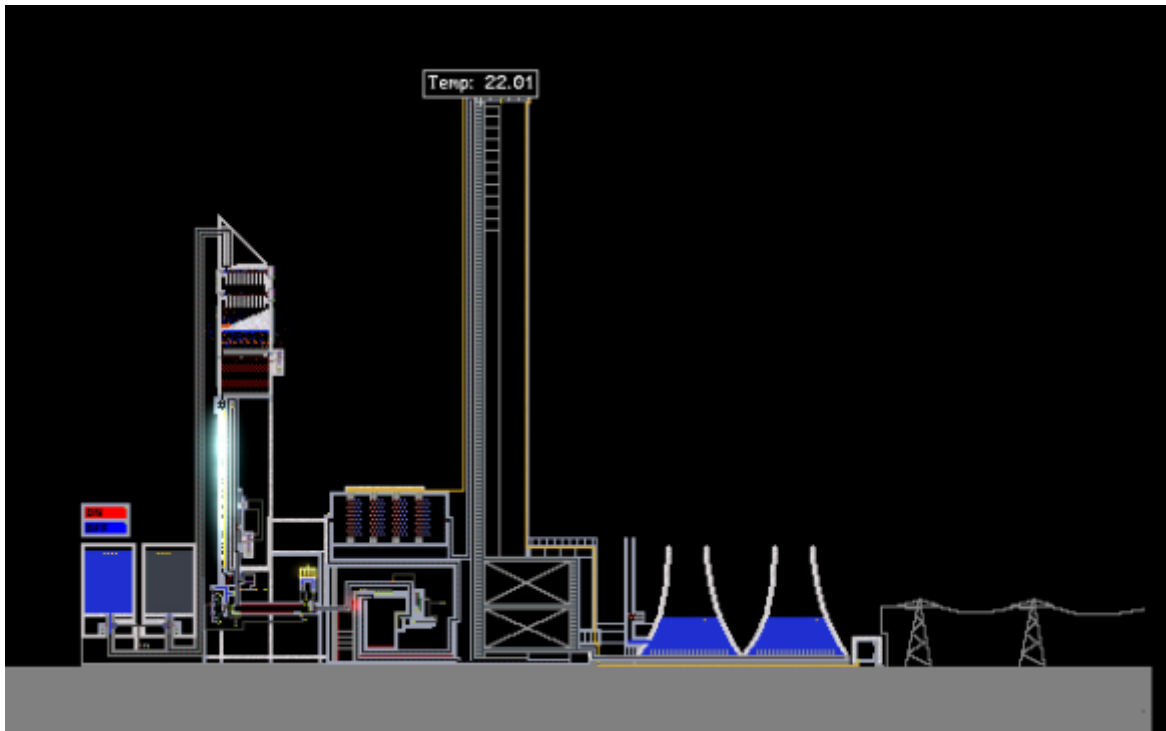
测试点编号	n, m	特殊性质
1 ~ 4	≤ 5	无
5 ~ 10	≤ 100	无
11 ~ 14	$\leq 2 \times 10^5$	$m = n - 1$
15 ~ 25	$\leq 2 \times 10^5$	无

TPT

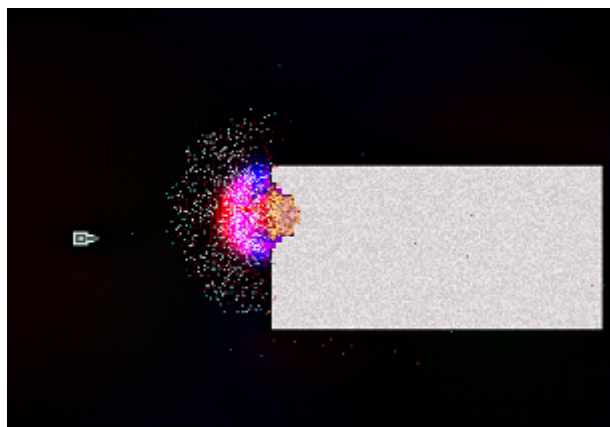
(tpt.cpp/.in/.out)

题目背景

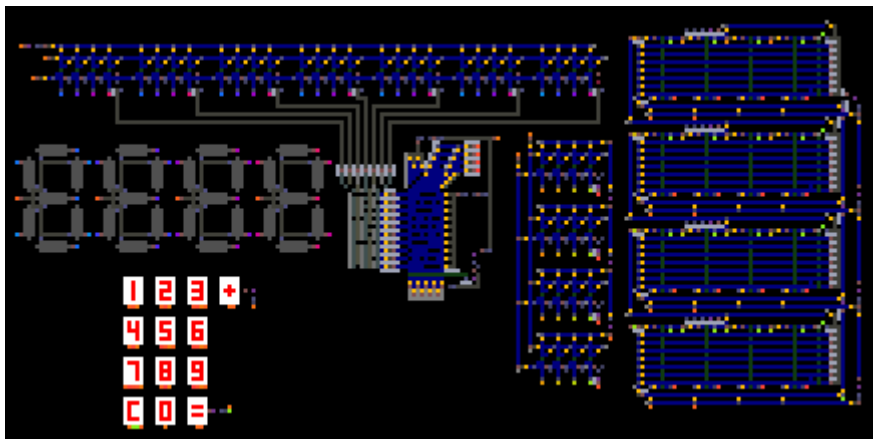
TPT 是一款经典的沙盒游戏，拥有非常真实的物理引擎。你可以在里面体验制造：



核电站



动能武器



计算器

等 **physical** 的装置的乐趣。你一定会喜欢它的。

题目描述

Sunny bunny 坐在火车上太无聊了，于是他打开了这款游戏。

现在他还只会摆弄最基础的元素之一——SAND。为了方便你理解，这里对 SAND 做出了一些修改。

游戏可以抽象成一个 $n \times m$ 的矩形网格，网格具有边界。如果一个格子为空，则该格子用 `.` 标注；如果一个格子里有 SAND，则该格子用 `#` 标注。

众所周知，受到重力作用的 SAND 会下落。不过，相邻的 SAND 会连结成一块，~~可能是因为 TPT 里的空气有大量水蒸气~~。具体来说，游戏里的 SAND 遵从如下的规则：

- 如果两个均有 SAND 的图格**四连通**，则这两个图格里的 SAND 连为一体。连接关系**具有传递性**；
- 极大的连为一体的 SAND 被看作一个物体；
- 从某个时刻开始，所有的物体会**同时开始以相同的速度匀速**下落（即向行增大的方向移动）。如果一个物体接触到了边界或者接触到了另外一个物体的顶部，则该物体会停止运动；在下落过程中，SAND **不会再连为一体**。
- 由于 SAND 之间的紧密粘合，**物体是刚体，总不会形变**。

如果对于 TPT 里面的简化物理定律还有疑惑，请参考**样例 1 解释**。

Sunny bunny 现在在网格里打开 `pause` 随意作画。他想知道，如果 `pause` 停止，也即所有物体同时开始运动，结束后网格变成什么样子呢？

输入格式

第一行输入两个正整数 n, m 。

接下来的 n 行，第 i 行包含一个长度为 m 的字符串，表示第 i 行的情况。

字符串仅会包含 `#` 和 `.` 两种字符，前者表示 SAND，后者表示空白图格。

输出格式

共输出 n 行，第 i 行应该包含一个长度为 m 的字符串，表示运动结束后，第 i 行的情况。
同样地，字符串仅应当包含 # 和 . 两种字符，前者表示 SAND，后者表示空白图格。

测试用例

样例 1 输入

```
5 4
..#.
##.#
.###
#...
#...
```

样例 1 输出

```
....
....
###.
###.
#..#
```

样例 1 解释

我们用不同的数字来标注不同的物体：

```
..2.
11.3
.11.
4...
4...
```

最终的局面就是：

```
....
....
112.
411.
4..3
```

样例 2 输入

```
8 8
...####.
...#....
...#.###
...#...#
...###.#
.....#
....####
.....
```

样例 2 输出

```
.....
.....
...####.
...#.###
...#...#
...#...#
...###.#
....####
```

样例 3

详见下发文件中的 tpt/tpt3.in 和 tpt/tpt3.out。

该样例满足测试点 1 的数据约定。

样例 4

详见下发文件中的 tpt/tpt4.in 和 tpt/tpt4.out。

该样例满足测试点 16 的数据约定。

数据规模与约定

本题采用子任务式测评。

对于 100% 的数据，满足 $1 \leq n \times m \leq 10^6$ 。

其余约定请见下表：

测试组编号	$n \times m$	特殊性质	分值
1	≤ 1000	不满足	30
2	$\leq 10^6$	满足	30
3	$\leq 10^6$	不满足	40

特殊性质：如果将所有的 SAND 看作图上的点，连为一体的 SAND 之间连接一条边。则图上的点度数要么为 0，要么为 2；且对于任何一个度数为 2 的连通块，它对应的图形都是凸的——即图形内部任意两点的连线都在图形内部。

元旦晚会

(party.cpp/.in/.out)

题目背景

时间的脚步是无声的，它在不经意间流逝，春去秋来，青春年华刹那方休。

题目描述

转眼间，2021 年便快要结束了。

Sunny bunny 正在为元旦晚会准备场地。在参会兔子们的自由分组之后，现在有 m 组兔子，每组均有 $2n$ 只兔子。每只兔子都有自己的身高。这不废话吗

Sunny bunny 需要为场地准备恰好 n 张双兔桌，且一开始就必须决定好准备哪些桌子，中途不可更换。桌子一共有 k 种，编号为 $1 \sim k$ 。第 i 张桌子对于兔子身高的合适区间为 $[L_i, R_i]$ 。当晚会开始的时候，每只兔子都会找桌子坐。因此，在每组的活动周期里，桌子总是恰好坐满的。

兔子们坐到了太高或者太矮的桌子都会感到不适，我们用不适指数来量化这个信息：

对于一只身高为 h 的兔子，如果他坐到了合适区间为 $[L, R]$ 的桌子，那么：

- 如果 $h < L$ ，则不适指数为 $L - h$ ；
- 如果 $L \leq h \leq R$ ，则不适指数为 0；
- 如果 $R < h$ ，则不适指数为 $h - R$ ；

兔子们会一组一组进入会场并落座，所有桌子都会坐两只兔子。两组兔子不会同时占用会场。

为了让兔子们能够度过一个愉快的夜晚，Sunny bunny 需要让 $2mn$ 只兔子的不适指数的总和最小。你能给出一个最小的结果作为参考吗？

输入格式

第一行输入三个正整数 m, n, k 。

接下来的 k 行，第 i 行输入 L_i, R_i 。

接下来的 m 行，每行包含 $2n$ 个正整数。将组编号 $1 \sim m$ ，每组的兔子们编号 $1 \sim 2n$ 后，第 i 行的第 j 个数 $h_{i,j}$ 表示第 i 组的第 j 只兔子的高度。

输出格式

输出一行一个非负整数，表示最小的不适指数之和。

测试用例

样例 1 输入

```
1 2 2
5 25
50 90
60 5 10 40
```

样例 1 输出

```
10
```

样例 1 解释

购买第 1 种桌子和第 2 种桌子各一张。

让第 2 只和第 3 只兔子坐第 1 种桌子，让第 1 只和第 4 只兔子坐第 2 种桌子。

总的不适指数之和为 $0 + 0 + 0 + 10 = 10$ 。

样例 2 输入

```
2 3 3
200 400
300 500
100 600
300 330 440 40 30 300
150 250 350 450 550 300
```

样例 2 输出

```
130
```

样例 3 输入

```
1 3 4
10 100
200 200
10 100
300 1000
5 10 20 15 200 90
```

样例 3 输出

```
105
```

样例 4

详见下发文件中的 party/party4.in 和 party/party4.out。

该样例满足测试点 11 的数据约定。

样例 5

详见下发文件中的 party/party5.in 和 party/party5.out。

该样例满足测试点 21 的数据约定。

样例 6

详见下发文件中的 party/party6.in 和 party/party6.out。

该样例满足测试点 26 的数据约定。

数据规模与约定

对于 100% 的数据，满足：

$1 \leq m, n \leq 2 \times 10^5; 1 \leq m \times n \leq 2 \times 10^5; 2 \leq k \leq 2 \times 10^5; 1 \leq L_i \leq R_i \leq 10^9; 1 \leq h_{i,j} \leq 10^9。$

其余约定请见下表：

测试点编号	m	n	k
1 ~ 5	≤ 100	$= 1$	≤ 50
6 ~ 10	$= 1$	≤ 1000	≤ 50
11 ~ 15	≤ 50	≤ 5	≤ 3
16 ~ 20	≤ 100	≤ 1000	$= 2$
21 ~ 25	≤ 100	≤ 1000	≤ 3
26 ~ 32	≤ 100	≤ 1000	≤ 50
33 ~ 37	≤ 100		
38 ~ 42		≤ 100	
43 ~ 50			

注：如果表格某一格留空，则表示该数据性质下，该列对应的信息没有特殊限制。