

NOI2017 模拟赛

时间：2017 年 6 月 23 日 08:00 ~ 13:00

题目名称	电报	三明治	回转寿司
题目类型	传统型	传统型	传统型
可执行文件名	telegraph.exe	sandwich.exe	sushi.exe
输入文件名	telegraph.in	sandwich.in	sushi.in
输出文件名	telegraph.out	sandwich.out	sushi.out
每个测试点时限	1.0 秒	2.0 秒	3.0 秒
内存限制	512 MB	512 MB	512 MB
测试点数目	10	20	25
每个测试点分值	10	5	4

提交源程序文件名

对于 C++ 语言	telegraph.cpp	sandwich.cpp	sushi.cpp
对于 C 语言	telegraph.c	sandwich.c	sushi.c
对于 Pascal 语言	telegraph.pas	sandwich.pas	sushi.pas

编译选项

对于 C++ 语言	-O2 -lm -Wl,--stack=536870912	-O2 -lm -Wl,--stack=536870912	-O2 -lm -Wl,--stack=536870912
对于 C 语言	-O2 -lm -Wl,--stack=536870912	-O2 -lm -Wl,--stack=536870912	-O2 -lm -Wl,--stack=536870912
对于 Pascal 语言	-O2	-O2	-O2

电报 (telegraph)

【题目背景】

某国由 n 座岛屿组成，每座岛上都有一台无线电接收机。第 i 座岛屿的无线电接收机可以接收来自第 a_i 座岛屿的信息。通过改变天线的方向，你可以改变第 i 座岛屿能够收到的信息的来源。改变第 i 座岛屿的无线电接收机的方向（即改变某个 a_i 的值）所需要的代价为 c_i 。

岛屿之间可以通过无线电来直接或间接地通信，即如果第 i 座岛屿能够发送信息给第 j 座岛屿，并且第 j 座岛屿能够发送信息给第 k 座岛屿，那么第 i 座岛屿就能够发送信息给第 k 座岛屿。

【题目描述】

现在你希望用最小的代价使得任意两座岛屿之间能够相互发送信息。

【输入格式】

从文件 *telegraph.in* 中读入数据。

输入第一行包含一个正整数 n ，表示岛屿的数量。

接下来 n 行，每行两个整数。第 i 行的两个整数为 a_i, c_i ，含义如题述。

【输出格式】

输出到文件 *telegraph.out* 中。

输出一行一个整数，表示最小代价。

【样例 1 输入】

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

【样例 1 输出】

```
4
```

【样例 2 输入】

4
2 2
1 6
1 3
3 1

【样例 2 输出】

5

【样例 3 输入】

4
2 2
1 3
4 2
3 3

【样例 3 输出】

4

【样例 4 输入】

3
2 1
3 1
1 1

【样例 4 输出】

0

【子任务】

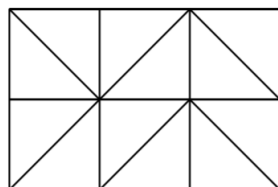
测试点	$n \leq$
1	10
2	15
3	
4	
5	3000
6	
7	
8	10^5
9	
10	

对于所有数据，保证 $2 \leq n \leq 10^5, 1 \leq a_i \leq n, a_i \neq i, 1 \leq c_i \leq 10^9$ 。

三明治 (sandwich)

【题目背景】

你订购了 $2nm$ 块三明治，并将它们摆成了一个 $n \times m$ 的矩形。



上图是 $n = 2, m = 3$ 的一种可能的情况，可以看到，每两块三明治构成了一个正方形。

你现在可以选择吃掉一些三明治。但为了避免麻烦，你希望每次吃的三明治都在边界上。也就是说，你~~不能~~吃掉一块三明治当且仅当以下两个条件同时成立

- 这块三明治的斜边与某块未被吃掉的三明治相邻；
- 这块三明治的至少一条直角边与某块未被吃掉的三明治相邻。

否则这块三明治可以被吃掉。

【题目描述】

现在你要求出，对于每个小正方形内的两块三明治，要把它们都吃掉，至少需要吃多少块三明治（包括这两块三明治本身）？

【输入格式】

从文件 *sandwich.in* 中读入数据。

输入第一行包含两个正整数 n, m ，表示行数和列数。

接下来 n 行，每行 m 个字符，其中第 i 行第 j 个字符为 N 表示这个正方形内的两块三明治的斜边沿主对角线放置，为 Z 表示沿副对角线放置。即



'N'



'Z'

【输出格式】

输出到文件 *sandwich.out* 中。

输出共 n 行，每行 m 个整数，第 i 行第 j 个整数表示吃掉位于第 i 行第 j 列的两块三明治，至少需要吃多少块三明治。

如果这是不可能的，那么请在对应位置输出 -1 。

【样例 1 输入】

```
2 3
NZN
ZZN
```

【样例 1 输出】

```
10 8 2
8 6 4
```

【样例 2 输入】

```
2 2
NZ
ZN
```

【样例 2 输出】

```
-1 -1
-1 -1
```

【样例 3 输入】

```
5 5
NZZZN
NNNZN
NNZNN
NZNNN
NZZZN
```

【样例 3 输出】

```
10 12 14 16 2
8 -1 -1 -1 4
6 -1 -1 -1 6
```

4 -1 -1 -1 8
2 16 14 12 10

【子任务】

测试点	$n, m \leq$	测试点	$n, m \leq$
1	50	11	400
2		12	
3		13	
4		14	
5		15	
6		16	
7		17	
8	400	18	
9		19	
10		20	

回转寿司 (sushi)

【题目背景】

你正在经营一家回转寿司店。店里有 n 位顾客围成了一个环，顾客的面前有一条沿逆时针旋转 ($1 \rightarrow 2 \rightarrow \dots \rightarrow n \rightarrow 1$) 的传送带。

每位顾客都有一碟寿司，每碟寿司都有一个价格。

现在有 q 碟按照时间顺序被送上来新寿司，第 i 碟寿司的价格为 p_i ，一开始它被放在第 s_i 位顾客面前的传送带上。接下来当每位顾客发现它面前的传送带上有一碟寿司时，他会将这碟寿司的价格与他现在拥有的寿司的价格相比较，如果传送带上的寿司的价格更便宜，那么他就会将自己的寿司与传送带上的寿司进行交换。

当这碟寿司从第 t_i 位顾客面前离开时，碟子会被立刻拿走。

【题目描述】

对于每一碟新寿司，你要求出这碟寿司在经过一系列的交换后，价格变成了多少。

【输入格式】

从文件 *sushi.in* 中读入数据。

第一行两个整数 n, q ，分别表示顾客数量与新寿司的数量。

接下来一行 n 个整数，第 i 个整数表示第 i 名顾客一开始的寿司的价格。

接下来 q 行，每行三个整数 s_i, t_i, p_i ，含义如题述。

【输出格式】

输出到文件 *sushi.out* 中。

输出共 q 行，每行一个整数，含义如题述。

【样例 1 输入】

```
6 7
8
6
7
4
5
9
2 4 5
```


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

【样例 1 输出】

7
9
8
7
8
6
5

【样例 2 输入】

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

【样例 2 输出】

7
5

【样例 3 输入】

10 10
19
5
8
17

14
3
9
10
7
6
1 8 4
7 3 2
5 9 10
4 8 3
10 3 6
8 7 4
6 6 3
2 9 12
6 3 7
9 6 3

【样例 3 输出】

19
10
14
17
8
10
3
12
7
9

【提示】

在第一组数据中，每次提供完新寿司后，顾客所拥有的寿司的价格依次为

1. 8, 5, 6, 4, 5, 9
2. 8, 5, 6, 4, 4, 5
3. 7, 5, 6, 4, 4, 5
4. 2, 5, 6, 4, 4, 5
5. 2, 5, 6, 4, 4, 5

6. 2, 5, 5, 1, 4, 47. 2, 5, 3, 1, 4, 4

【子任务】

测试点	$n \leq$	$q \leq$	其他约束
1	2000	2000	无
2			
3			
4	400000	25000	$s_i = 1$ 且 $t_i = n$
5			
6			
7			
8			
9			
10			
11			无
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
25			

对于所有数据，保证 $1 \leq n \leq 400000, 1 \leq q \leq 25000, 1 \leq x_i, p_i \leq 10^9, 1 \leq s_i, t_i \leq n$ 。