

2018 Spring Round

ljs

2018 年 3 月 5 日

时间：5 小时

题目名称	生日聚会	排水管	Vim
题目类型	传统型	传统型	传统型
目录	brunhilda	pipes	vim
输入文件名	brunhilda.in	pipes.in	vim.in
输出文件名	brunhilda.out	pipes.out	vim.out
每个测试点时限	1s	1s	1s
内存限制	256MB	128MB	512MB
测试点数目	10	N/A	25
每个测试点分值	10	N/A	4

提交程序文件名

对于 C++ 语言	brunhilda.cpp	pipes.cpp	vim.cpp
对于 C 语言	brunhilda.c	pipes.c	vim.c
对于 Pascal 语言	brunhilda.pas	pipes.pas	vim.pas

编译选项

对于 C++ 语言	-lm,-O2	-lm,-O2	-lm,-O2
对于 C 语言	-lm,-O2	-lm,-O2	-lm,-O2
对于 Pascal 语言	-lm,-O2	-lm,-O2	-lm,-O2

注意：1、请确保在选手目录下直接存放有源程序。

2、对于 Subtask 数据，你只有通过一个 Subtask 中的所有子测试点才可以获得相应的分数，请注意特殊情况，尽量保证算法的正确性。

生日聚会

(brunhilda.cpp/c/pas)

【题目背景】

Brunhilda 就快要 7 岁了。今年她决定在拉脱维亚过生日。拉脱维亚位于东欧波罗的海沿岸，有欧洲北部巴黎之称。首都里加建筑美轮美奂，而远离首都的小镇更是自然风光无限好，休闲时间她也去野外、海边、山地露营游玩。现在，她正在筹备她的 7 岁生日 party。

【问题描述】

Brunhilda 发明了如下的游戏：所有的小朋友围成一圈跑动起来，直到听到数字 k 。接着，所有的小朋友都互相抱紧尽量使自己在一个人数正好为 k 的小组中。只要剩下的小朋友个数至少有 k 个，那么就一定会形成一个新的 k 个小朋友的小组。最后，会留下少于 k 个小朋友，那么这些小朋友就会被淘汰。出于 Brunhilda 对骑士盔甲的喜爱，最后所有的小朋友都会被淘汰。一旦小朋友全部被淘汰了，这轮游戏也就结束了。

Brunhilda 叫她的爸爸 Wotan 帮忙报数字。Wotan 并没有表现出对这个游戏多大的喜爱，于是在一开始就报出了 ∞ 。Brunhilda 觉得在 party 上出现这样的状况实在是太尴尬了，于是她给了她的爸爸一些质数，要他每次报数时从中间选择一个数字。爸爸可以使用同一个数字多次。

Wotan 想要尽快的结束这个游戏，因为他还要坐飞机去德国看 FC Asgard 俱乐部的足球赛。现在有 Q 轮游戏， Q 轮参与游戏的人数分别为 n_1, \dots, n_Q ，Wotan 想提前知道对于每一轮游戏 $n_j (1 \leq j \leq Q)$ ，至少需要报出几个数字才可以结束这轮游戏（即淘汰全部的 n_j 个小朋友）。

【输入格式】

从文件 *brunhilda.in* 中读入数据。

输入文件第一行包括两个整数 m 和 Q 。

第二行包含 m 个质数，表示 Wotan 可以使用的质数。接下来 Q 行每行一个整数 $n_j (1 \leq j \leq Q)$ ，表示这一轮游戏的人数。

【输出格式】

输出到文件 *brunhilda.out* 中。

输出文件包括 Q 行, 对于一个 $n_j (1 \leq j \leq Q)$, 如果 Wotan 可以结束游戏, 那么输出报出的数字个数; 否则输出字符串 `oo` (两个小写字母 'o', 表示 $+\infty$)。

【样例输入 1】

```
2 2
2 3
5
6
```

【样例输出 1】

```
3
oo
```

【样例 1 解释】

第一轮 5 个小朋友做游戏。

第一次 Wotan 可以报出 3, 那么淘汰 2 名小朋友, 剩下 3 名小朋友继续游戏;

第二次 Wotan 可以报出 2, 那么淘汰 1 名小朋友, 剩下 2 名小朋友继续游戏;

第三次 Wotan 可以报出 3, 那么淘汰 2 名小朋友, 所有小朋友均被淘汰, 第一轮游戏结束。

第二轮游戏, 无论怎么报数都不可以淘汰所有的 6 个小朋友, 所以无法结束游戏。

【样例输入/输出 2】

见 `down/brunhilda` 下的 `brunhilda2.in/brunhilda2.ans`。

【子任务】

对于 20% 的数据, $m, n_j, Q \leq 10000$ 。

对于另外 20% 的数据, $Q = 1$ 。

对于 100% 的数据, $1 \leq m \leq 100000, 1 \leq Q \leq 100000, 2 \leq p_i \leq 10000000, 1 \leq n_j \leq 10000000$ 。

排水管

(pipes.cpp/c/pas)

【题目背景】

Hotham 城再次被头号罪犯 Jester 攻击了！

【问题描述】

这一次，Jester 的攻击目标是 Hotham 城的自来水系统。Hotham 城的自来水存储在 $N * M$ 个水库中。这些水库形成了一个 $N * M$ 的网格图。相邻的两个水库有排水管连接（包括上下左右斜对角 8 个方向），从任意一个水库一定可以到达其相邻的另一个水库。另外每一个水管一定连接两个不同的水库。而每一对水库也一定只存在最多一个水管连接它们。

Jester 在若干个水库中分别投放了一个污染源，现在他坐在控制室施行破坏行动。每一次，他会选择一条管道，假设这条管道连接的两个水库为 u, v ，那么：

- 如果 u, v 均没有污染源，则不会产生任何影响
- 如果 u 有污染源， v 没有污染源，则会对城市产生 1 的污染，相应的位于 u 的污染源会流至 v
- 如果 v 有污染源， u 没有污染源，则会对城市产生 1 的污染，相应的位于 v 的污染源会流至 u
- 如果 u 有污染源， v 也有污染源，则会对城市产生 2 的污染， u, v 的污染源会交换

Jester 在进行若干次破坏后逃走了。

Hotham 的市长在每个水库设置了勘测器，勘测器可以测出污染源在**与这个水库相邻的所有水管流动过程中产生的污染**。不过，这些勘测器中有些机器出现了不同程度的故障，它们会不定时的**额外增加测量的污染值**，这也就意味着就算 Jester 没有进行任何操作，一些勘测器显示的污染值仍会大于零。注意，每次的增加值**不必相同**。市长没有在水管处设置勘测器，因此，他只能观察到每一个水库的情况，他不知道每个水管内是如何流动的，更不知道它们一共流动变化了多少次。勘测器还记录下了 Jester 开始破坏前与破坏结束时每个水库处是否有污染源。

你的任务是编写一个程序帮助市长。你将得到开始与结束时整个自来水中每个水库有无污染源的具体情况以及在这个水库的勘测器测量到的污染值。不过市长**并不能保证给你的数据准确无误**，它们可能在传输的过程中部分发生了改变。你的程序应该判断由所给数据是否可以确定 Jester 给城市带来的最小污染。注意，任何时候一个水库最多只会含有一个污染源。如果可以，请输出城市受到的最小的污染值。

【输入格式】

从文件 *pipes.in* 中读入数据。

输入的第一行包含两个数字 N, M ($1 \leq N \leq 20, 1 \leq M \leq 20$)，表示网格图的大小。接下来一个 $N * M$ 的 01 矩阵，表示每一个水库开始时有无污染源。接下来一个 $N * M$ 的 01 矩阵，表示每一个水库结束时有无污染源。接下来一个 $N * M$ 的矩阵（为 0-9 的数字或字母 'z'，z 表示 74），表示每一个水库处的勘测器测量到的污染值。

【输出格式】

输出到文件 *pipes.out* 中。

如果你认为你得到数据不能使你确定最小值，请输出-1。否则，输出一个数，表示城市至少受到的最小的污染。

【样例输入 1】

```
3 3
110
000
001
000
110
100
222
222
222
```

【样例输出 1】

```
4
```

【样例 1 解释】

城市所受到的所有可能的污染值中最小的为 4，即城市至少受到了 4 的污染。

【样例输入/输出 2】

见 *down/pipes* 下的 *pipes2.in/pipes2.ans*。

【子任务】

Subtask 1(20%): $1 \leq N \leq 3, 1 \leq M \leq 3$

Subtask 2(9%): 开始时最多只有 1 个水库处有污染源

Subtask 3(11%): 开始时最多只有 2 个水库处有污染源

Subtask 4(9%): 开始时最多只有 3 个水库处有污染源

Subtask 5(15%): 你所得到的每个勘测器测量到的污染值为 74
(用字母‘z’输入), 且开始时 1 的个数不超过 30

Subtask 6(12%): $1 \leq N \leq 12, 1 \leq M \leq 12$

Subtask 7(24%): $1 \leq N \leq 20, 1 \leq M \leq 20$

Vim

(vim.cpp/c/pas)

【题目背景】

Ernest Vincent Wright (1872 –October 7, 1939[1]) was an American author known for his book *Gadsby*, a 50,000-word novel which, except for the introduction and a note at the end, did not use the letter "e". ——Wikipedia

【问题描述】

Victor 是 Ernest 的狂热粉丝，他尝试学习 Ernest 写小说，但他给了自己更大的挑战。他只用字母表的前 10 个字母进行写作 (abcdefghij)。无奈的是，在写作中途，键盘的 ‘e’ 键坏了。为了保持一致性，他决定删去所有他已经写下的 ‘e’ 字母。他的一个程序员朋友建议他用文本编辑器 Vim 来完成这项工作。不幸的是，Victor 并不擅长使用 Vim，他只会 Vim 的三个最基本的操作 ‘x’, ‘h’, ‘f’。

- ‘x’ 操作会将当前光标处的字母删除，并且光标的位置（从左数）不会发生改变。值得注意的是，Victor 并不会在光标移动到文件最末时使用这项操作。
- ‘h’ 操作会将光标的位置前移一步（向左）。值得注意的是，当光标处于第一个字母时，这项操作不会产生任何影响。
- ‘f’ 操作时，之后会紧接一个字母 *C*，这项操作会将光标移动到下一个字母 *C* 出现的位置（即使光标所处的位置也正好是字母 *C*）。当然，如果光标的右侧并没有 *C* 字母出现，那么什么也不会发生。

例如，当前文本是：

jeffiehadabigidea

如果用下划线表示光标的位置，那么：

- 操作 ‘x’ 后文本将变成 jeffehadabigidea
- 操作 ‘h’ 后文本将变成 jeffiehadabigidea
- 操作 ‘fi’ 后文本将变成 jeffiehadabigidea

你的任务是写一个程序计算 Victor 将所有文本中的 ‘e’ 字母删去的最小按键次数。注意，你并不能将文本中的其他字母删去。最开始时，光标位于文本第一个字母处。另外，Victor 还好心的提醒你，由于键盘的 ‘e’ 键坏了，所有的 ‘fe’ 操作是不被接受的。

【输入格式】

从文件 *vim.in* 中读入数据。

输入的第一行包含一个整数 N ，表示文本的总长度。接下来一行描述这个文本，数据保证每一个字母都会在 ‘a’ 到 ‘j’ 的范围内。保证文本的第一个和最后一个字母不是 ‘e’。

【输出格式】

输出到文件 *vim.out* 中。

输出一个整数，表示将文件中所有的 ‘e’ 删除的最小按键次数。

【样例输入 1】

```
5
aebec
```

【样例输出 1】

```
7
```

【样例 1 解释】

一个可行的最优方案是：

```
fchxhxx
```

你可以通过 vim 自行测试。参考方案：打开终端输入 `vim file.txt` 以打开 `file.txt`，按下 `:q <ENTER>` 退出（退出前请保存好文件，即按下 `:w <ENTER>`）。

【样例输入 2】

```
35
chefeddiefedjeffeachbigagedegghad
```


【样例输出 2】

36

【样例 2 解释】

一个可行的最优方案是：

fdhxhhx f f h x f a h x h h x h h x f d h x f g h x f a h x

【样例输入/输出 3】

见 *down/vim* 下的 *vim3.in/vim3.ans*。

【子任务】

对于 100% 的数据, $N \leq 70000$ 。

每个测试点还满足如下约束：

测试点编号	N	特殊性质
1	≤ 70000	没有字母 ‘e’
2	≤ 5	无
3-6	≤ 20	
7-13	≤ 500	
14-15	≤ 5000	
16-25	≤ 70000	