第 33 届全国信息学奥林匹克竞赛

CCF NOI 2016

第一试

竞赛时间：2016 年 7 月 24 日 8:30-13:30

题目名称	优秀的拆分	网格	循环之美
目录	excellent	grid	cyclic
可执行文件名	excellent	grid	cyclic
输入文件名	excellent.in	grid.in	cyclic.in
输出文件名	excellent.out	grid.out	cyclic.out
每个测试点时限	1.5 秒	2 秒	2 秒
内存限制	512MB	1GB	512MB
测试点数目	20	25	25
每个测试点分值	5	4	4
是否有部分分	否	否	否
题目类型	传统型	传统型	传统型
是否有样例文件	是	是	是
是否有附加文件	否	否	否

提交源程序须加后缀

对于 C++ 语言	excellent.cpp	grid.cpp	cyclic.cpp
对于 C 语言	excellent.c	grid.c	cyclic.c
对于 Pascal 语言	excellent.pas	grid.pas	cyclic.pas

编译开关

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

优秀的拆分

【问题描述】

如果一个字符串可以被拆分为 $AABB$ 的形式，其中 A 和 B 是任意**非空**字符串，则我们称该字符串的这种拆分是优秀的。

例如，对于字符串 aabaabaa，如果令 $A=$ aab，$B=$ a，我们就找到了这个字符串拆分成 $AABB$ 的一种方式。

一个字符串可能没有优秀的拆分，也可能存在不止一种优秀的拆分。比如我们令 $A=$ a，$B=$ baa，也可以用 $AABB$ 表示出上述字符串；但是，字符串 abaabaa 就没有优秀的拆分。

现在给出一个长度为 n 的字符串 S，我们需要求出，在它**所有子串**的所有拆分方式中，优秀拆分的总个数。这里的子串是指字符串中**连续**的一段。

以下事项需要注意：

1. 出现在不同位置的相同子串，我们认为是不同的子串，它们的优秀拆分均会被计入答案。
2. 在一个拆分中，允许出现 $A=B$。例如 cccc 存在拆分 $A=B=$ c。
3. 字符串本身也是它的一个子串。

【输入格式】

输入文件为 ***excellent.in***。

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。保证 $1 \le T \le 10$。

接下来 T 行，每行包含一个仅由英文小写字母构成的字符串 S，意义如题所述。

【输出格式】

输出文件为 ***excellent.out***。

输出 T 行，每行包含一个整数，表示字符串 S 所有子串的所有拆分中，总共有多少个是优秀的拆分。

【样例 1 输入】

```
4
aabbbb
cccccc
aabaabaabaa
bbaabaababaaba
```

【样例 1 输出】

```
3
5
4
7
```

【样例 1 说明】

我们用 $S[i,j]$ 表示字符串 S 第 i 个字符到第 j 个字符的子串（从 1 开始计数）。

第一组数据中，共有 3 个子串存在优秀的拆分：

$S[1,4] =$ aabb，优秀的拆分为 $A =$ a，$B =$ b；

$S[3,6] =$ bbbb，优秀的拆分为 $A =$ b，$B =$ b；

$S[1,6] =$ aabbbb，优秀的拆分为 $A =$ a，$B =$ bb。

而剩下的子串不存在优秀的拆分，所以第一组数据的答案是 3。

第二组数据中，有两类，总共 4 个子串存在优秀的拆分：

对于子串 $S[1,4] = S[2,5] = S[3,6] =$ cccc，它们优秀的拆分相同，均为 $A =$ c，$B =$ c，但由于这些子串位置不同，因此要计算 3 次；

对于子串 $S[1,6] =$ cccccc，它优秀的拆分有 2 种：$A =$ c，$B =$ cc 和 $A =$ cc，$B =$ c，它们是相同子串的不同拆分，也都要计入答案。

所以第二组数据的答案是 $3 + 2 = 5$。

第三组数据中，$S[1,8]$ 和 $S[4,11]$ 各有 2 种优秀的拆分，其中 $S[1,8]$ 是问题描述中的例子，所以答案是 $2 + 2 = 4$。

第四组数据中，$S[1,4]$，$S[6,11]$，$S[7,12]$，$S[2,11]$，$S[1,8]$ 各有 1 种优秀的拆分，$S[3,14]$ 有 2 种优秀的拆分，所以答案是 $5 + 2 = 7$。

【样例 2 输入输出】

见选手目录下的 ***excellent/excellent2.in*** 与 ***excellent/excellent2.ans***。

【样例 3 输入输出】

见选手目录下的 ***excellent/excellent3.in*** 与 ***excellent/excellent3.ans***。

【子任务】

对于全部的测试点，保证 $1 \leq T \leq 10$。以下对数据的限制均是对于单组输入数据而言的，也就是说同一个测试点下的 T 组数据均满足限制条件。

我们假定 n 为字符串 S 的长度，每个测试点的详细数据范围见下表：

测试点编号	n	其他约束
1、2	≤ 300	S 中所有字符全部相同
3、4	≤ 2000	
5、6	≤ 10	无
7、8	≤ 20	
9、10	≤ 30	
11、12	≤ 50	
13、14	≤ 100	
15	≤ 200	
16	≤ 300	
17	≤ 500	
18	≤ 1000	
19	≤ 2000	
20	≤ 30000	

网格

【问题描述】

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（$0 \le c \le nm$），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（0 个，1 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：我们用图表示一只跳蚤，用图表示一只蛐蛐，那么图 1 描述了一个 $n = 4, m = 4, c = 2$ 的情况。

这种情况下蛐蛐国王可以通过将第 2 行第 2 列，和第 3 行第 3 列的两只跳蚤替换为蛐蛐，从而达成他的希望，如图 2 所示。并且，不存在更优的方案，但是可能存在其他替换 2 只跳蚤的方案。

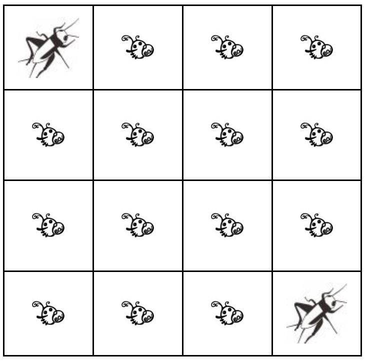

图 1

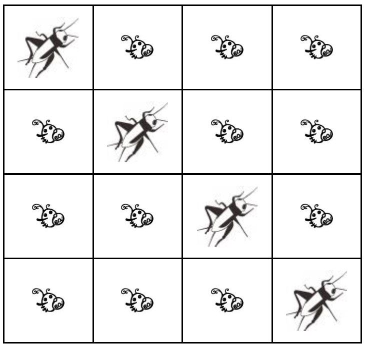

图 2

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】

输入文件为 ***grid.in***。

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。保证 $1 \le T \le 20$。

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。保证 $1 \le n, m \le 10^9$，$0 \le c \le \min(nm, 10^5)$。

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（$1 \le x \le n, 1 \le y \le m$）。每一组数据当中，同一个蛐蛐不会被多次描述。

同一行相邻的整数之间由一个空格隔开。

【输出格式】

输出文件为 ***grid.out***。

对于每一组数据依次输出一行答案。

如果这组数据中，蛐蛐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 输入】

```
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 1 输出】

```
2
1
0
-1
```

【样例 1 说明】

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 2 行第 2 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**。

【样例 2 输入输出】

见选手目录下的 ***grid/grid2.in*** 与 ***grid/grid2.ans***。

【提示】

如果你的程序需要用到较大的栈空间（这通常意味着需要较深层数的递归），请务必仔细阅读选手目录下的文档 ***stack.pdf***，以了解在最终评测时栈空间的限制与在当前工作环境下调整栈空间限制的方法。

【子任务】

对于全部的测试点，保证 $1 \le T \le 20$。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，$\sum c \le 10^5$。

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

每个测试点的详细数据范围见下表。表中的 n, m, c 均是对于单个输入数据（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 $\sum c$ 是对于单个测试点而言的。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。

<table>
<tr><th colspan="2">n,m</th><th>测试点</th><th>c</th></tr>
<tr><td colspan="2">$nm \le 4$</td><td>1</td><td rowspan="7">$c \le nm$</td></tr>
<tr><td colspan="2">$nm \le 8$</td><td>2</td></tr>
<tr><td colspan="2">$nm \le 15$</td><td>3</td></tr>
<tr><td colspan="2">$nm \le 30$</td><td>4</td></tr>
<tr><td colspan="2">$nm \le 100$</td><td>5</td></tr>
<tr><td colspan="2">$nm \le 300$</td><td>6</td></tr>
<tr><td colspan="2">$nm \le 1000$</td><td>7</td></tr>
<tr><td colspan="2" rowspan="3">$nm \le 20000$</td><td>8</td><td>$c \le 5$</td></tr>
<tr><td>9</td><td>$c \le 15$</td></tr>
<tr><td>10</td><td>$c \le 30$</td></tr>
<tr><td rowspan="5">$n,m \le 20000$</td><td>$nm \le 20000$</td><td>11</td><td rowspan="5">$\sum c \le 20000$</td></tr>
<tr><td>$nm \le 10^5$</td><td>12</td></tr>
<tr><td>$nm \le 3 \times 10^5$</td><td>13</td></tr>
<tr><td>$nm \le 10^6$</td><td>14</td></tr>
<tr><td>$nm \le 10^9$</td><td>15</td></tr>
<tr><td colspan="2">$n,m \le 10^5$</td><td>16</td><td>$\sum c \le 10^5$</td></tr>
<tr><td colspan="2" rowspan="9">$n,m \le 10^9$</td><td>17</td><td>$c = 0$</td></tr>
<tr><td>18</td><td>$c \le 1$</td></tr>
<tr><td>19</td><td>$c \le 2$</td></tr>
<tr><td>20</td><td>$c \le 3$</td></tr>
<tr><td>21</td><td>$c \le 10$</td></tr>
<tr><td>22</td><td>$c \le 30$</td></tr>
<tr><td>23</td><td>$c \le 300$</td></tr>
<tr><td>24</td><td>$\sum c \le 20000$</td></tr>
<tr><td>25</td><td>$\sum c \le 10^5$</td></tr>
</table>

循环之美

【问题描述】

牛牛是一个热爱算法设计的高中生。在他设计的算法中，常常会使用带小数的数进行计算。牛牛认为，如果在 k 进制下，一个数的小数部分是**纯循环**的，那么它就是美的。

现在，牛牛想知道：对于已知的十进制数 n 和 m，在 k 进制下，有多少个数值上**互不相等**的纯循环小数，可以用分数 $\frac{x}{y}$ 表示，其中 $1 \le x \le n$，$1 \le y \le m$，且 x, y 是整数。

一个数是纯循环的，当且仅当其可以写成以下形式：

$$a.\dot{c_1}c_2c_3 \ldots c_{p-1}\dot{c_p}$$

其中，a 是一个整数，$p \ge 1$；对于 $1 \le i \le p$，c_i 是 k 进制下的一位数字。

例如，在十进制下，$0.45454545\ldots\ldots = 0.\dot{4}\dot{5}$ 是纯循环的，它可以用 $\frac{5}{11}$、$\frac{10}{22}$ 等分数表示；在十进制下，$0.1666666\ldots\ldots = 0.1\dot{6}$ 则不是纯循环的，它可以用 $\frac{1}{6}$ 等分数表示。

需要特别注意的是，我们认为一个整数是纯循环的，因为它的小数部分可以表示成 0 的循环或是 $k-1$ 的循环；而一个小数部分非零的有限小数不是纯循环的。

【输入格式】

输入文件为 ***cyclic.in***。

输入文件只有一行，包含三个十进制正整数 n, m, k，意义如题所述。

【输出格式】

输出文件为 ***cyclic.out***。

只输出一行一个整数，表示满足条件的美的数的个数。

【样例 1 输入】

```
2 6 10
```

【样例 1 输出】

```
4
```

【样例 1 说明】

满足条件的数分别是：

$1/1 = 1.0000 \dots \dots$

$1/3 = 0.3333 \dots \dots$

$2/1 = 2.0000 \dots \dots$

$2/3 = 0.6666 \dots \dots$

`1/1` 和 `2/2` 虽然都是纯循环小数，但因为它们相等，因此只计数一次；同样，`1/3` 和 `2/6` 也只计数一次。

【样例 2 输入】

```
23333 666666 310
```

【样例 2 输出】

```
5089564081
```

【提示】

这部分将提供一个将分数化为对应的小数的方法，如果你已经熟悉这个方法，你不必阅读本提示。

分数可以通过除法，用分子除以分母化为对应的小数。有些分数在除法过程中无法除尽，这样的分数在不断进行的除法过程中余数一定会重复出现。从商数的个位所对应的余数起，设第一次重复出现的余数前两次出现的位置所对应的商数位分别是小数点后第 a 位和小数点后第 b 位（特殊地：如果其中一个对应的商数位是个位，则认为 $a = 0$；不妨设 $a < b$），则其循环部分可以用小数点后第 $a + 1$ 位到小数点后第 b 位的循环来表示。

例如：在十进制下，将 $\frac{5}{11}$ 转化为小数时，个位开始的商数依次为 $4,5,4,\dots$，对应的余数分别为 $6,5,6,\dots$。余数第一次重复出现的位置是个位和小数点后第 2 位，那么 $a = 0, b = 2$ 即其循环部分可以用小数点第 1 位到第 3 位来表示。

表示为：$\frac{5}{11} = 0.45454545\ldots = 0.\dot{4}\dot{5}$。

在十进制下，将 $\frac{1}{6}$ 转化为小数时，个位开始的商数依次为 1,6,6, …，对应的余数分别为 4,4,4 …。余数第一次重复出现的位置是小数点后第 1 位和小数点后第 2 位，即其循环部分可以用小数点后第 2 位来表示。表示为：$\frac{1}{6} = 0.1666\ldots\ldots = 0.1\dot{6}$。

需要注意的是：**商数**重复出现并不代表进入了循环节。

【子任务】

对于所有的测试点，保证 $1 \le n \le 10^9, 1 \le m \le 10^9, 2 \le k \le 2000$。对于每个测试点，有以下约束（其中留空的表示没有特殊的约束）：

测试点编号	n	m	k	测试点编号	n	m	k
1	≤ 10	≤ 20	$= 2$	13	$\le 10^5$	$\le 10^8$	≤ 100
2	≤ 100	$\le 10^4$	$= 2$	14	$\le 2 \times 10^5$		≤ 1000
3	≤ 1000		$= 2$	15	$\le 5 \times 10^5$		
4	≤ 10000		$= 2$	16	$\le 10^6$	$\le 10^8$	≤ 100
5	≤ 10	≤ 20	$= 3$	17	$\le 2 \times 10^6$		≤ 1000
6	≤ 100	$\le 10^4$	$= 3$	18	$\le 5 \times 10^6$		
7	≤ 1000		$= 3$	19	$\le 10^7$	$\le 10^8$	≤ 100
8	≤ 10000		$= 3$	20	$\le 2 \times 10^7$		≤ 1000
9	≤ 10	≤ 20	≤ 100	21	$\le 2 \times 10^7$		
10	≤ 100	$\le 10^4$	≤ 100	22	$\le 10^8$	$\le 10^8$	
11	≤ 1000		≤ 1000	23	$\le 10^8$	$\le 10^8$	
12	≤ 10000			24、25			