

A 栈与队列

时空限制：5s 128MB

文件名

queue.in/queue.out/queue.cpp

题目描述

给出 N 个已经塞了数进去的栈（每个栈中元素的数量可能不同），有一个空的「答案队列」，你每次可以「将一个栈的栈顶元素弹出，插入答案队列的末尾」，直至所有栈都清空。试求「字典序最小」的答案队列。

如果两个答案队列 a, b （从队首往队尾数）前 $i - 1$ 个数都相同，而 $a_i < b_i$ ，则我们称 a 的字典序比 b 的字典序小。

输入格式

第一行一个整数 N 。

接下来 N 行，每行第一个整数为 L ，表示栈中元素的数量。接下来按照从栈顶到栈底的顺序依次给出 L 个整数。

输出格式

$\sum L$ 个整数，表示字典序最小的答案队列。

样例 #1

样例输入 #1

```
3
1 2
1 100
1 1
```

样例输出 #1

```
1 2 100
```

样例 #2

样例输入 #2

```
2
5 10 20 30 40 50
2 28 27
```

样例输出 #2

```
10 20 28 27 30 40 50
```

样例 #3

样例输入 #3

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

样例输出 #3

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

提示

对于 40% 的数据，保证， $1 \leq N \leq 10, 1 \leq L \leq 10$ 。

对于 100% 的数据，保证， $1 \leq N \leq 1000, 1 \leq L \leq 1000$ 。

B 前缀树

时空限制：2s 64MB

文件名

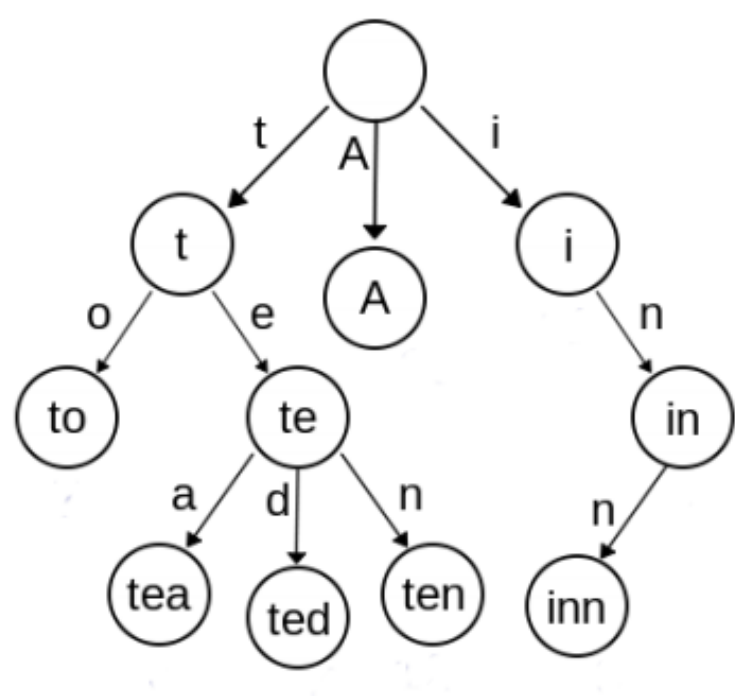
trie.in/trie.out/trie.cpp

题目描述

Matej 面临着一个难题。在此之前，我们必须熟悉一种称作前缀树（`trie`）的数据结构。前缀树以前缀的方式，储存单词：

- 前缀树的每一条边都用英文字母表中的字母表示。
- 前缀树的根节点表示空前缀。
- 前缀树的每个其他节点都表示一个非空前缀。依次连接根节点至该节点路径上所标有的字母，即可得到该前缀。
- 不存在从一个节点出发的、标有相同字母的两条边。

例如，这棵前缀树储存了 `A,to,tea,tet,ten,i,in,inn`：



现在，Matej 获得了 n 个单词，并可以将其中的一些单词重组。例如 `abc` 可以重组为 `acb,bac,bca,cab,cba`。请你计算，将一些单词重组后，储存这些单词的前缀树节点数的最小值。

输入格式

第一行一个整数 n 。

接下来 n 行，每行一个字符串，表示 Matej 获得的单词。

输出格式

一行，一个整数，表示将一些单词重组后，储存这些单词的前缀树节点数的最小值。

样例 #1

样例输入 #1

```
3
a
ab
abc
```

样例输出 #1

```
4
```

样例 #2

样例输入 #2

```
3
a
ab
c
```

样例输出 #2

```
4
```

样例 #3

样例输入 #3

```
4
baab
abab
aabb
bbaa
```

样例输出 #3

```
5
```

提示

样例 3 解释

所有单词均可以重组为 `aabb`。显然，前缀树最少的节点数应为 5（包含了表示空前缀的根节点）。

数据规模与约定

对于 100% 的数据，保证 $1 \leq n \leq 16$ 。

所有单词的长度和不超过 10^6 ，且只包含小写字母。

C 绘图程序

时空限制：1s 128MB

文件名

slika.in/slika.out/slika.cpp

题目描述

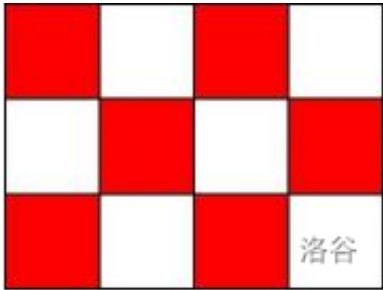
Mirko 刚刚安装了一个全新的绘图程序。该程序支持 K 种不同的颜色，用从 1 到 K 的整数表示。所有的绘图都是在一个尺寸为 $N \times N$ 的画布上完成的。在开始时，所有格子都是白色的(用 1 表示)。

画布左上角的单元格为坐标 $(0, 0)$ 。第一个坐标 x 表示行，第二个坐标 y 表示列。

Mirko 最喜欢的消遣是使用 `PAINT c x1 y1 x2 y2` 命令绘制矩形棋盘图案，其中 c 表示所选的颜色， $(x1, y1)$ 和 $(x2, y2)$ 分别是左上的坐标和右下的坐标。

矩形左上角的单元格将被绘制为所选的颜色，而其余的则如棋盘一样涂色。没有被所选颜色覆盖的细胞将保持它们之前的颜色。

例如，一个白色的画布涂上一个红色的棋盘图案看起来就像这样的：



Mirko最近发现了另外两个命令。他可以随时保存他的画通过使用 `SAVE` 的命令，稍后使用 `load x` 命令再次加载它，其中 x 表示保存的序列号的正整数。

不幸的是，程序崩溃了，Mirko 的画永远丢失了。幸运的是，Mirko 用了个保存所有已使用命令的日志。你能帮 Mirko 修复那幅丢失的画吗？

输入格式

输入的第一行包含三个正整数 N, K, M ， N 代表画布的边长， K 代表有 K 种颜色， M 代表命令个数。

下面的 M 行每一行都包含描述的三个命令中的一个。输入将不包含任何非法的命令。

输出格式

输出共 N 行，每一行包含 N 个表示单元格颜色的整数，对应一行的画。

样例 #1

样例输入 #1

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

样例输出 #1

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

样例 #2

样例输入 #2

```
3 3 4
PAINT 3 0 0 1 1
SAVE
PAINT 2 1 1 2 2
LOAD 1
```

样例输出 #2

```
3 1 1
1 3 1
1 1 1
```

样例 #3

样例输入 #3

```
3 4 7
PAINT 2 0 0 1 1
SAVE
PAINT 3 1 1 2 2
SAVE
PAINT 4 0 2 0 2
LOAD 2
PAINT 4 2 0 2 0
```

样例输出 #3

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

提示

【样例解释#1】

命令 1 将 $(0,0)$ 到 $(3,3)$ 的格子染成了棋盘式，即把 $(0,0)$, $(0,2)$, $(1,1)$, $(1,3)$, $(2,0)$, $(2,2)$, $(3,1)$, $(3,3)$ 都染成了 2。

命令 2 将 $(0,3)$ 到 $(3,3)$ 的格子染成了棋盘式，即把 $(0,3)$, $(2,3)$ 染成了 3。

【数据范围】

对于 30% 的数据， $1 \leq N \leq 100$, $1 \leq M \leq 3500$ 。

对于 100% 的数据， $1 \leq N \leq 1000$, $2 \leq K \leq 10^5$, $1 \leq M \leq 10^5$ 。

D 两次

时空限制：5s 512MB

文件名

poklon.in/poklon.out/poklon.cpp

题目描述

给定一个包含 N 个自然数的数组。

接着需要回答 Q 次询问，每次询问输出区间 $[L, R]$ 内恰好出现两次的自然数的数量。

输入格式

第一行，两个整数 N, Q ，分别表示数组元素数量和询问次数。

第二行， N 个整数，表示数组中的元素。

接下来的 Q 行，每行两个整数 L, R ，表示询问的区间。

输出格式

共 Q 行，依次对应每次询问的结果。

样例 #1

样例输入 #1

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

样例输出 #1

```
1
```

样例 #2

样例输入 #2

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

样例输出 #2

```
0
1
```

样例 #3

样例输入 #3

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

样例输出 #3

```
0
2
```

提示

【样例 1 解释】

区间 $[1, 3]$ 中只有 1 恰好出现了两次。

【数据规模与约定】

对于 40% 的数据， $N, Q \leq 5000$ 。

对于 100% 的数据， $1 \leq N, Q \leq 5 \times 10^5$ ， $1 \leq L \leq R \leq N$ ，数组中的元素都是小于 10^9 的自然数。