CCF CSP 计算机软件能力认证

CCF CSP

第 33 次认证

时间：2024 年 3 月 31 日 13:30 ~ 17:30

题目名称	词频统计	相似度计算	化学方程式配平	十滴水	文件夹合并
题目类型	传统型	传统型	传统型	传统型	传统型
输入	标准输入	标准输入	标准输入	标准输入	标准输入
输出	标准输出	标准输出	标准输出	标准输出	标准输出
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	3.0 秒	2.0 秒
内存限制	512 MiB	512 MiB	512 MiB	512 MiB	512 MiB
子任务数目	10	10	10	7	9
测试点是否等分	是	是	是	否	否

词频统计（tfidf）

【题目描述】

在学习了文本处理后，小 P 对英语书中的 n 篇文章进行了初步整理。具体来说，小 P 将所有的英文单词都转化为了整数编号。假设这 n 篇文章中共出现了 m 个不同的单词，则把它们从 1 到 m 进行编号。这样，每篇文章就简化为了一个整数序列，其中每个数都在 1 到 m 范围内。

现给出小 P 处理后的 n 篇文章，对于每个单词 i（$1 \leq i \leq m$），试统计：

1. 单词 i 出现在了多少篇文章中？
2. 单词 i 在全部文章中总共出现了几次？

【输入格式】

从标准输入读入数据。

输入共 $n+1$ 行。

输入的第一行包含两个正整数 n 和 m，分别表示文章篇数和单词编号上限。

输入的第 $i+1$ 行（$1 \leq i \leq n$）包含由空格分隔的若干整数，其中第一个整数 l_i 表示第 i 篇文章的长度（单词个数）；接下来 l_i 个整数表示对应的整数序列，序列中每个整数均在 1 到 m 范围内，各对应原文中的一个单词。

【输出格式】

输出到标准输出。

输出共 m 行。

第 i 行（$1 \leq i \leq m$）输出由空格分隔的两个整数 x_i 和 y_i，表示共有 x_i 篇文章包含单词 i，总计出现次数为 y_i。

【样例输入】

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

【样例输出】

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

【样例解释】

单词 2 在：

- 文章 1 中出现两次；
- 文章 3 中出现三次；
- 文章 4 中出现一次。

因此 $x_2 = 3$、$y_2 = 6$。

【子任务】

全部的测试数据满足 $0 < n, m \leq 100$，且每篇文章至少包含一个单词、最多不超过 100 个单词（$1 \leq l_i \leq 100$）。

相似度计算（jaccard）

【题目背景】

两个集合的 Jaccard 相似度定义为：

$$Sim(A,B)=\frac{|A\cap B|}{|A\cup B|}$$

即交集的大小除以并集的大小。当集合 A 和 B 完全相同时，$Sim(A,B)=1$ 取得最大值；当二者交集为空时，$Sim(A,B)=0$ 取得最小值。

【题目描述】

除了进行简单的词频统计，小 P 还希望使用 Jaccard 相似度来评估两篇文章的相似性。具体来说，每篇文章均由若干个英文单词组成，且英文单词仅包含“大小写英文字母”。对于给定的两篇文章，小 P 首先需要提取出两者的单词集合 A 和 B，即去掉各自重复的单词。然后计算出：

- $|A\cap B|$，即有多少个不同的单词同时出现在两篇文章中；
- $|A\cup B|$，即两篇文章一共包含了多少个不同的单词。

最后再将两者相除即可算出相似度。需要注意，在整个计算过程中应当**忽略英文字母大小写**的区别，比如 **the**、**The** 和 **THE** 三者都应被视作同一个单词。

试编写程序帮助小 P 完成前两步，计算出 $|A\cap B|$ 和 $|A\cup B|$；小 P 将亲自完成最后一步的除法运算。

【输入格式】

从标准输入读入数据。

输入共三行。

输入的第一行包含两个正整数 n 和 m，分别表示两篇文章的单词个数。

第二行包含空格分隔的 n 个单词，表示第一篇文章；

第三行包含空格分隔的 m 个单词，表示第二篇文章。

【输出格式】

输出到标准输出。

输出共两行。

第一行输出一个整数 $|A\cap B|$，即有多少个不同的单词同时出现在两篇文章中；

第二行输出一个整数 $|A\cup B|$，即两篇文章一共包含了多少个不同的单词。

【样例 1 输入】

```
3 2
The tHe thE
the THE
```

【样例 1 输出】

```
1
1
```

【样例 1 解释】

$A = B = A \cap B = A \cup B =$ **{the}**

【样例 2 输入】

```
9 7
Par les soirs bleus dete jirai dans les sentiers
PICOTE PAR LES BLES FOULER LHERBE MENUE
```

【样例 2 输出】

```
2
13
```

【样例 2 解释】

$A =$ **{bleus, dans, dete, jirai, les, par, sentiers, soirs}** $|A| = 8$
$B =$ **{bles, fouler, les, lherbe, menue, par, picote}** $|B| = 7$
$A \cap B =$ **{les, par}** $|A \cap B| = 2$

【样例 3 输入】

```
15 15
Thou that art now the worlds fresh ornament And only herald to the
   gaudy spring
```

```
Shall I compare thee to a summers day Thou art more lovely and
   more temperate
```

【样例 3 输出】

```
4
24
```

【子任务】

80% 的测试数据满足：$n, m \leq 100$ 且所有字母均为小写；
全部的测试数据满足：$n, m \leq 10^4$ 且每个单词最多包含 10 个字母。

化学方程式配平（balancing）

【题目背景】

近日来，西西艾弗岛化学研究中心的研究员们向岛上的初中学生开展了化学科普活动。在活动中发现，初学化学的同学们十分苦恼于正确配平化学方程式。而还有一些同学，则提出了一些稀奇古怪的方程式，让研究员们帮忙配平。在配平之前，研究员们需要先判断这个方程式是否能够配平。

一个化学方程式，也叫化学反应方程式，是用化学式表示化学反应的式子。其等号左右两侧分别列举了化学反应的全部反应物和生成物。每种物质都用其化学式表示。一个物质的化学式，列举了构成该物质的各元素的原子数目。例如，水的化学式是 H_2O，表示水分子中含有两个氢原子和一个氧原子。化学方程式中每种物质的化学式前面都有一个系数，表示参与反应或生成的物质的相对数目比例。例如，方程式 $2H_2 + O_2 = 2H_2O$ 表示二分子氢气和一分子氧气反应生成二分子水。我们称一个化学方程式是配平的，是指该方程式中的反应物和生成物中，各元素原子总数目相等。例如上述方程式中，左侧氢原子、氧原子的总数目分别为 4 和 2，右侧氢原子、氧原子的总数目分别为 4 和 2，因此该方程式是配平的。

【题目描述】

为了配平一个化学方程式，我们可以令方程式中各物质的系数为未知数，然后针对涉及的每一种元素，列出关于系数的方程，形成一个齐次线性方程组。然后求解这个方程组，得到各物质的系数。这样，我们就把化学方程式配平的问题，转化为了求解齐次线性方程组的问题。如果方程组没有非零解，那么这个方程式是不可以配平的。反之，如果方程组有非零解，我们就可能得到一个配平的方程式。当然，最终得到的方程式仍然需要结合化学知识进行检验，对此我们不再进一步考虑，仅考虑非零解的存在。

例如要配平化学方程式：$Al_2(SO_4)_3 + NH_3 \cdot H_2O \to Al(OH)_3 + (NH_4)_2SO_4$

首先假定所有物质在方程的同一侧，即不考虑哪个是反应物，哪个是生成物，分别设这些物质的系数为 x_1, x_2, x_3, x_4，则可以针对出现的各个元素，列出如下的方程组：

$$
\begin{aligned}
2x_1 + 0x_2 + x_3 + 0x_4 &= 0 \quad \text{Al} \\
3x_1 + 0x_2 + 0x_3 + x_4 &= 0 \quad \text{S} \\
12x_1 + x_2 + 3x_3 + 4x_4 &= 0 \quad \text{O} \\
0x_1 + x_2 + 0x_3 + 2x_4 &= 0 \quad \text{N} \\
0x_1 + 5x_2 + 3x_3 + 8x_4 &= 0 \quad \text{H}
\end{aligned}
$$

用矩阵的形式表示为：

$$\begin{pmatrix} 2 & 0 & 1 & 0 \\ 3 & 0 & 0 & 1 \\ 12 & 1 & 3 & 4 \\ 0 & 1 & 0 & 2 \\ 0 & 5 & 3 & 8 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \mathbf{0}$$

对系数矩阵实施高斯消元，得到系数矩阵的一个行阶梯形式：

$$\begin{pmatrix} 2 & 0 & 1 & 0 \\ 0 & 1 & -3 & 4 \\ 0 & 0 & -\frac{3}{2} & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

由此可见，系数矩阵的秩为 3。根据线性代数的知识，我们知道，齐次线性方程组 $\mathbf{AX} = \mathbf{0}$ 的解空间的维数等于其未知数个数减去系数矩阵的秩 rank$\mathbf{A}$。而要让方程式配平，即要求方程组存在非零解，那么就需要让解空间的维数大于 0，即系数矩阵的秩小于未知数个数。因此，我们可以通过判断系数矩阵的秩是否小于未知数个数，来判断方程式是否可以配平。如果可以配平，则可以通过解的符号来判断反应物和生成物的位置。

本题中，我们将给出一些化学方程式，请你按照上述方法判断它们是否可以配平。为了便于程序处理，我们用到的化学式，会被化简为只包含小写字母和数字的字符串，不包含括号。其中连续的字母表示一种元素，随后的数字表示原子个数。原子个数为 1 时不省略数字；一个化学式中包含的元素不重复。例如，上述方程式中的化学式可以化简为 **al2s3o12**、**n1h5o1**、**al1o3h3**、**n2h8s1o4**。

【输入格式】

从标准输入读入数据。

输入的第一行包含一个正整数 n，表示需要判断的化学方程式的个数。

接下来的 n 行，每行描述了一个需要被配平的化学方程式。包含空格分隔的一个正整数和全部涉及物质的化学式。其中，正整数 m 表示方程式中的物质；随后的 m 个字符串，依次给出方程式中的反应物的化学式和生成物的化学式。

【输出格式】

输出到标准输出。

输出包含 n 行，每行包含字母 **Y** 或 **N**，表示按题设方法，所给待配平化学方程式能否配平。

【样例 1 输入】

```
6
2 o2 o3
3 c1o1 c1o2 o2
2 n2o4 n1o2
4 cu1 h1n1o3 cu1n2o6 h2o1
4 al2s3o12 n1h5o1 al1o3h3 n2h8s1o4
4 c1o1 c1o2 o2 h2o1
```

【样例 1 输出】

```
Y
Y
Y
N
Y
Y
```

【样例 1 解释】

输入中给出了 5 个待配平的化学方程式，其中各方程式的配平情况为：

- $3O_2 = 2O_3$
- $2CO + O_2 = 2CO_2$
- $N_2O_4 = 2NO_2$
- 因为缺少生成物 NO 或 NO_2，所以不可以配平
- $Al_2(SO_4)_3 + 6NH_3 \cdot H_2O = 2Al(OH)_3 + 3(NH_4)_2SO_4$
- $2CO + O_2 = 2CO_2$，本方程式对应的线性方程组求解后，得到 H_2O 的系数为 0，说明其未参与反应，属多余的物质。在这种情况下，由于对应的线性方程组存在非零解，所以我们仍然认为这个方程式是可以配平的。

【子任务】

对于 20% 的数据，每个方程中物质的个数不超过 2，每个方程中涉及的全部元素不超过 2 种；

对于 60% 的数据，每个方程中物质的个数不超过 3，每个方程中涉及的全部元素不超过 3 种；

对于 100% 的数据，每个方程中物质的个数不超过 40，每个方程中涉及的全部元素不超过 40 种；且有 $1 \leq n \leq 10$，且化学式中各元素的原子个数不超过 50。

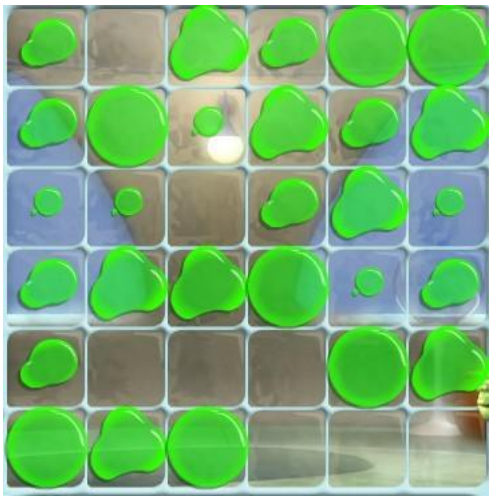
【提示】

- 对矩阵进行高斯消元的一种方法是：
    1. 考察矩阵的第一列上的元素：
        - 若全都为零，则对除去该列的子矩阵重复上述判断；
        - 若不全为零，则：
            1. 考察第一行第一列的元素：
                * 如果其为 0，则将该行与后面的某一个第一列非 0 的行交换，使第一行第一列的元素非 0；
            2. 令后续所有行减去第一行的适当倍数，使得后续所有行的第一列元素为 0；
    2. 对除去第一行第一列的子矩阵重复上述操作，直至不再余下子矩阵。
- 对系数矩阵高斯消元后，不全为 0 的行的数目即为系数矩阵的秩。
- 评测环境仅提供各语言的标准库，特别地，不提供任何线性代数库。

十滴水（tendrop）

【题目描述】

十滴水是一个非常经典的小游戏。



小 C 正在玩一个一维版本的十滴水游戏。我们通过一个例子描述游戏的基本规则。

游戏在一个 $1 \times c$ 的网格上进行，格子用整数 $x(1 \le x \le c)$ 编号，编号从左往右依次递增。网格内 m 个格子里有 $1 \sim 4$ 滴水，其余格子里没有水。在我们的例子中，$c = m = 5$，按照编号顺序，每个格子中分别有 $2, 4, 4, 4, 2$ 滴水。

玩家可以进行若干次操作，每次操作中，玩家选择一个有水的格子，将格子的水滴数加一。任何时刻若某个格子的水滴数大于等于 5，这个格子里的水滴就会向两侧**爆开**。此时，这个格子的水被清空，同时对于左方、右方两个方向**同时**进行以下操作：找到当前格子在对应方向上最近的有水的格子，如果存在这样的格子，将这个格子的水滴数加一。**若在某个时刻，有多个格子的水滴数大于等于 5，则最靠左的先爆开。**

在我们的例子中，若玩家对第三格进行操作，则其水滴数变为 5，故第三格水滴爆开，水被清空，其左侧最近的有水格子（第二格）和右侧最近的有水格子（第四格）的水量增加 1，此时每个格子中分别有 $2, 5, 0, 5, 2$ 滴水。

此时第二格和第四格的水滴数均大于等于 5，按照规则，第二格的水先爆开，爆开后每个格子中分别有 $3, 0, 0, 6, 2$ 滴水；最后第四格的水滴爆开，每个格子中分别有 $4, 0, 0, 0, 3$ 滴水。

小 C 开始了一局游戏并进行了 n 次操作。小 C 在每次操作后，会等到所有水滴数大于等于 5 的格子里的水滴都爆开再进行下一次操作。

小 C 想知道他的水平有多高，于是他想知道每一次操作后还有多少格子里有水。

保证这 n 次操作都是合法的，即每次操作时操作的格子里都有水。

【输入格式】

从标准输入读入数据。

输入的第一行三个整数 c, m, n 分别表示网格宽度、有水的格子个数以及操作次数。
接下来 m 行每行两个整数 x, w，表示第 x 格有 w 滴水。
接下来 n 行每行一个整数 p，表示小 C 对第 p 格做了一次操作。

【输出格式】

输出到标准输出。
输出 n 行，每行一个整数表示这次操作之后网格上有水的格子数量。

【样例 1 输入】

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

【样例 1 输出】

```
2
1
```

【子任务】

对于所有测试数据，

- $1 \le c \le 10^9$，$1 \le m \le \min(c, 3 \times 10^5)$，$1 \le n \le 4m$；
- $1 \le x, p \le c$，$1 \le w \le 4$；
- 输入的所有 x 两两不同；
- 对于每个输入的 p，保证在对应操作时 p 内有水。

<table>
<tr><th>子任务编号</th><th>$c \leq$</th><th>$m \leq$</th><th>特殊性质</th><th>分值</th></tr>
<tr><td>1</td><td>30</td><td>30</td><td rowspan="2">有</td><td rowspan="2">15</td></tr>
<tr><td>2</td><td rowspan="2">$3,000$</td><td rowspan="3">$3,000$</td></tr>
<tr><td>3</td><td rowspan="3">无</td><td>10</td></tr>
<tr><td>4</td><td>10^9</td><td rowspan="4">15</td></tr>
<tr><td>5</td><td>3×10^5</td><td rowspan="3">3×10^5</td></tr>
<tr><td>6</td><td rowspan="2">10^9</td><td>有</td></tr>
<tr><td>7</td><td>无</td></tr>
</table>

特殊性质：在游戏的任意时刻（包括水滴爆开的连锁反应过程中），只有至多一个格子的水滴数大于等于 5。

文件夹合并（merge）

【题目描述】

新入职西西艾弗岛有限公司的小 C 接替了刚刚升职的小 S 的项目。然而小 C 打开项目工程时，一层层嵌套的文件夹让小 C 感到眼花缭乱。为了精简项目结构，小 C 决定对项目的文件夹进行一些必要的合并。

项目中共有 n 个文件夹。为了方便，我们用 1 至 n 的整数给这 n 个文件夹编号，其中编号为 1 的文件夹为项目的根文件夹，其他每个文件夹都有一个父文件夹，这些文件夹构成了树形结构。除了子文件夹以外，第 i 个文件夹内还直接存储了 d_i 字节的数据。

小 C 进行了若干次文件夹合并操作。每次操作中小 C 会选择一个文件夹 x_j，将这个文件夹和它的所有子文件夹合并。具体地，小 C 会进行以下操作：遍历 x_j 的子文件夹 y，将文件夹 y 包含的所有文件夹和文件移动到文件夹 x_j，然后删除文件夹 y。所有文件和文件夹的名称是两两不同的，合并过程中不需要考虑文件或文件夹重名的情况。在每一次合并操作后，小 C 需要知道文件夹 x_j 内共有几个文件夹以及多少字节的数据。

例如，考虑以下项目：根文件夹内有文件夹 2 和文件夹 3 以及 100 字节数据，其中文件夹 2 为空文件夹，文件夹 3 内有 200 字节数据和文件夹 4，文件夹 4 包含 300 字节数据。对根文件夹进行一次合并后，文件夹 2 和文件夹 3 被合并至根文件夹，此时根文件夹下有文件夹 4 以及 300 字节数据，而文件夹 4 下也包含 300 字节数据。

在合并文件夹的过程中，小 C 常常需要访问某个文件夹 z_j 下的文件。此时，小 C 会从根文件夹开始，每次进入当前文件夹的一个子文件夹。小 C 需要知道按照以上过程，获取到文件夹 z_j 下的文件至少需要经过多少个文件夹。

例如，在以上项目中，未对根文件夹进行合并前，访问根文件夹下的文件只需要经过根文件夹一个文件夹，而访问文件夹 4 则需要经过根文件夹以及文件夹 3 和 4。而对根文件夹进行合并之后，访问文件夹 4 只需要经过根文件夹和文件夹 4 了。

在整个项目中，小 C 一共进行了 m 次文件夹合并以及文件访问操作。你需要帮助小 C 正确维护文件夹之间的关系，并在每次操作后正确回答小 C 需要的数据。

【输入格式】

从标准输入读入数据。

输入的第一行两个整数 n, m，分别表示文件夹数量以及操作次数。

第二行 $(n-1)$ 个整数 $f_2, \cdots, f_n$，其中 f_i 表示文件夹 i 的父文件夹编号。

第三行 n 个整数 $d_1, d_2, \cdots, d_n$，其中 d_i 表示文件夹 i 中数据的存储量。

接下来 m 行第 j 行两个整数，第一个整数 op_j 表示操作类型。若 $op_j = 1$ 则表示一次文件夹合并操作，接下来一个整数 x_j 表示合并的文件夹编号；若 $op_j = 2$ 则表示一次文件访问操作，接下来一个整数 z_j 表示访问的文件夹编号。

【输出格式】

输出到标准输出。

输出 m 行，第 j 行表示第 j 个操作中小 C 需要的数据：若 $op_j = 1$ 则输出两个整数，依次表示文件夹 x_j 的子文件夹数量以及数据的存储量；若 $op_j = 2$ 则输出一个整数表示小 C 获取文件夹 z_j 下的数据最少需要经过的文件夹个数。

【样例 1 输入】

```
4 6
1 1 3
100 0 200 300
2 1
2 4
1 1
2 4
1 1
1 1
```

【样例 1 输出】

```
1
3
1 300
2
0 600
0 600
```

【子任务】

对于所有测试数据，

- $1 \le n \le 5 \times 10^5, 1 \le m \le 3 \times n$，
- $1 \le f_i \le n$，输入的文件夹结构构成树形结构，
- $0 \le d_i \le 10^5$，
- $1 \le x_j, z_j \le n$，每次合并操作中给出的文件夹 x_j 没有被删除，每次文件访问操作中给出的文件夹 z_j 没有被删除。

子任务编号	$n \leq$	特殊性质	分值
1	500	无	10
2	$5,000$		15
3	10^5		
4	5×10^5	A	5
5		B	
6		C	10
7		D	15
8		E	10
9		无	15

特殊性质 A：$f_i = (i-1)$。

特殊性质 B：$f_i = 1$。

特殊性质 C：在文件夹合并操作中，$x_j = 1$。

特殊性质 D：$op_j = 1$，即没有文件访问操作。

特殊性质 E：$op_j = 2$，即没有文件夹合并操作。