语音识别

【题目背景】

与机器进行语音交流，让机器明白你说什么，这是人们长期以来梦寐以求的事情。语音识别技术就是让机器通过识别和理解过程把语音信号转变为相应的文本或命令的高技术。

——百度百科

现在，我们需要你解决的是一个简化版的语音识别问题：

麦克风所录入的信息可以被认为是一些独立的信号，每个信号都按照其电平值被表示为一个非负整数，这些信号组成的有序序列就是麦克风输入的信号序列。

语音信号实例

信号序列可以用一个非负整数序列来描述，形如 $A = \{a_1, a_2, \ldots a_n\}$，信号序列 A 的子序列 A'是指 A 中的一段连续信号 $A' = \{a_i, a_{i+1}, \ldots a_{j-1}, a_j\}$。

实际情况中，麦克风录入的信号序列往往混有为数不多的噪声，为了在语音识别中能够处理噪声带来的问题，需要引入近似匹配的概念：

设 A、B 是两个信号序列，A 对于 B 近似匹配，是指从 A 中删除若干个信号之后，所得的信号序列恰好等于 B。我们把从 A 中删除信号的个数称为 A 对于 B 近似匹配的差别程度。值得注意的是，为了使得识别有意义，只有那些差别程度不大的近似识别才是有意义的。例如，从信号序列{1, 2, 0}中删除一个信号 2 就可以得到信号序列{1, 0}，因此{1, 2, 0}对于{1, 0}近似匹配，差别程度为 1；同样{1, 2, 0}对于{0}也是近似匹配的，其差别程度为 2，但是如果限定近似匹配的差别程度不能超过 1，那么{1, 2, 0}对于{0}的近似匹配就将被忽略。特别的，如果两个信号序列完全一致，那么这两个信号序列的匹配可以被认为是差别程度为 0 的近似匹配。

研究人员已经对很多日常使用的字进行了预处理，得到了和每个字相对应的信号序列，这些字的信号序列所组成的集合称为字典。令 A‘是信号序列 A 的一个子序列，如果 A‘对于信号序列 B 近似匹配，那么 A‘就是 B 在 A 中的一次近似出现。例如：

对于信号序列 $A = \{a_1, a_2, a_3, a_4, a_5, a_6\} = \{3, 3, 3, 1, 2, 0\}$，那么 $A_1 = \{a_1, a_2, a_3\}$, $A_2 = \{a_4, a_5\}$, $A_3 = \{a_6\}$, $A_4 = \{a_1, a_2\}$都是 A 的子序列。

考虑字典∑ = {a = {1}, b = {1, 2}, c = {0}, d = {3, 3}}，且近似匹配的差别程度上限为 1，那么 A_1、A_4 都是 d 在 A 中的一次近似出现，A_2 是 a 或者 b 的近似出现，A_3 是 c 的近似出现。

进一步可知字典中的每一个字都可以在信号序列中被检测出来。

a 被检测出 3 次，$\{a_4\}$,$\{a_3, a_4\}$,$\{a_4, a_5\}$；

b 被检测出 3 次，$\{a_4, a_5\}$, $\{a_3, a_4, a_5\}$, $\{a_4, a_5, a_6\}$；

c 被检测出 2 次，$\{a_6\}$, $\{a_5, a_6\}$；

d 被检测出 4 次，$\{a_1, a_2\}$, $\{a_2, a_3\}$, $\{a_1, a_2, a_3\}$, $\{a_2, a_3, a_4\}$。

为了对麦克风输入的信号序列作出尽量好的识别，一个最直观的想法是，从输入信号序列中能够识别出来的字越多越好。具体的说，如果能够从信号序列 $A = \{a_1, a_2, \ldots a_n\}$中找出一组子序列 $D_1, D_2, \ldots D_s$，对于字典∑，满足：

1) 对于任意 i、j，D_i 和 D_j 没有相交部分；也就是说，如果 $D_i = \{a_p, a_{p+1}, \ldots a_q\}$，$D_j = \{a_u, a_{u+1}, \ldots a_v\}$，那么一定有区间[p, q]和区间[u, v]没有交集，即 $p > v$ 或者 $q < u$。
2) 任意 D_i 都可以在字典中找到一个字(设为 $word_i$）的信号序列，使得 D_i 对于字 $word_i$ 的信号序列近似匹配；

那么 D_1->$word_1$, D_2->$word_2$, ... D_s->$word_s$ 被称为 A 的一种识别方案，把这些近似匹配按照出现先后排列起来就可得到识别结果，s 就是这种识别方案的长度。最直观的想法就是希望能够找出一组最长的识别方案。

例如，考虑刚才的例子：{A_1->d, A_3->c}, {A_4->d, A_2->a, A_3->c}，{A_4->d, A_2->b, A_3->c}都是可行的识别方案，对应的识别结果分别是："dc", "dac", "dbc"。由于不存在长度超过 3 的识别方案，所以最长的识别方案的长度为 3，有两种识别结果"dac"和"dbc"。

【任务】

给定信号序列 *A*，字典∑，和能够允许的近似匹配的差别程度的上限 *K*。要求计算：

- 字典中有多少字在 *A* 中近似出现，总的近似出现的次数是多少？
- *A* 的最长的识别方案的长度是多少，在识别方案长度最长的前提下，能够识别出的本质不同的序列有多少个？

注意：如果两个识别方案分别对应识别结果 $words = \{word_1, word_2, \ldots word_p\}$ 和 $words' = \{word'_1, word'_2, \ldots word'_q\}$，如果 words 和 words'不完全相同，则两个识别方案本质不同。

【输入文件】

第一行有三个整数 *N*、*M*、*K*，分别表示待测信号序列的长度、字典的大小和近似匹配差别程度的上限。

第二行有 *N* 个用空格隔开的非负整数，描述待测信号序列 *A*。

接下来 M 行每行描述字典的一个不同的字，对于其中每一行：

先输入一个非负整数 L，表示这个字对应信号序列的长度，接下来 L 个整数给出这个字所对应的信号序列。相邻的整数之间用空格隔开。

注意：不同的字可能对应相同或者类似的信号序列。

【输出文件】

输出文件一共包含两行：

第一行包含两个整数，分别表示字典中可以被检测出来的字的个数，以及近似出现的总次数；

第二行包含两个整数，分别表示最长识别方案的长度，以及能够识别出的本质不同的最长识别方案数（输出的方案数为你所得到的答案对 1 000 003 取模后的结果）。

注意：每行要求包含且仅包含两个整数，用一个空格隔开，不允许多余的空格和换行，不允许输出不完整。以下的输出均不合法（其中下划线表示空格，\n 表示回车）：

多空格： 100__100\n 100_100\n	多回车： 100_100\n 100_100\n \n	多行首（尾）空格 _100_100\n 100_100_\n	输出不完整 100_100\n 100\n

【样例输入】

```
6 4 1
3 3 3 1 2 0
1 1
2 1 2
1 0
2 3 3
```

【样例输出】

```
4 12
3 2
```

【样例说明】

样例为在题目背景中给出的例子。

【评分标准】

每个测试点单独评分。

如果输出文件不存在、不合法则该测试点得 0 分，否则，你的得分为以下 4 项得分之和：

能够识别的字的个数正确(2 分)

近似出现的总次数正确(4 分)

最长识别方案的长度正确(2 分)

最长识别方案的个数正确（2 分）

【数据约定】

本题共有 10 个测试点，每个测试点 10 分，每个测试点单独评分

Test#	$N \le$	$M \le$	$K \le$	$L \le$	*Test#*	$N \le$	$M \le$	$K \le$	$L \le$
1	6	5	1	10	6	100000	20	2	15000
2	1000	10	5	100	7	100000	20	20	100000
3	50000	15	5	50000	8	100000	20	20	50000
4	10000	20	20	15000	9	100000	20	20	100000
5	50000	20	20	30000	10	100000	20	20	100000

其中 L 表示字典中所有字的信号总长度。