

诡异的文本处理

【问题描述】

伟大的人类智慧之神 clevertick 为了处理他测试机器人时使用的数据，制造了一个文本处理器，命名为 CTEX (CleverTick's Editing eXecutor)。

CTEX 的模型由若干个结点和结点之间的一些转移(可以视为有向边)组成。模型中的所有结点有且只有一个是初始结点，并且至少有一个是终止结点，初始结点不可能是终止结点。

在某些结点上可能存在输出装置，输出装置的类型用 0 到 $S - 1$ 的整数表示。类型为 i 的输出装置可以在指针(关于 CTEX 的指针会在后面描述)到达该装置所在的结点时，在所处理的纸带末尾写上一个数 i 。

除终止结点外，每个结点都至少有一个从该结点出发的转移，上面写有一个 0 到 $S - 1$ 的整数。CTEX 模型保证不存在两个从同一个结点出发且写有相同数值的转移。

有一个待处理的文本以一个非负整数序列的形式记录在一条纸带上(之后所有的读写操作都在这条纸带上完成)，序列中所有整数的范围为 0 到 $S - 1$ 。处理器有一个最开始指在初始结点上的指针。

当 CTEX 启动时，若初始结点上存在输出装置，则在纸带的末尾写上一个表示该输出装置类型的数。之后会重复以下步骤直到结束：

处理器首先判断目前指针指的结点是否为终止结点。若为终止结点，则打印目前纸带上的所有内容，处理器正常结束。若不是终止结点，如果目前纸带为空(没有任何数值了)，处理器会非正常结束(输出错误信息然后结束)。若纸带不为空，则会进行如下操作：

从纸带开头读取一个数，设为 i ，同时将这个数从纸带上删去。之后处理器会查找是否存在从指针指的结点出发且写有 i 的转移，若存在，则指针指到该转移到达的结点；若不存在写有 i 的转移，则指针的位置不会变化(即停留在原结点)。在完成结点转移操作后，若此时指针所在的结点上存在输出装置，则在纸带末尾写上一个表示该输出装置的类型的数，若不存在输出装置则什么也不写。

clevertick 用 CTEX 处理了很多文本，大大方便了他测试机器人的工作。不幸的是，反人类智慧者最终没有放过 CTEX，某天，clevertick 发现 CTEX 被破坏了一——只剩下结点和转移，转移上写的数值和结点上的输出装置均已丢失。

clevertick 决定使用以前记录的信息恢复 CTEX。他以前记录的信息包括他曾经用 CTEX 处理过哪些文本以及它们对应的输出结果(纸带内容或者非正常结束信息)，此外他还可以从以前的记录中估算每个转移上写了各个数、每个结点上的各类型输出装置存在的概率大小。为了表示这些概率，他给了每个概率一个权值，该权值与对应概率取对数后的值的相反数成正比。关于这个权值，我们有一个推论：对于一个 CTEX 恢复方案，所有的需要恢复的信息(转移上写的数值和输出装置类型)对应权值之和越小就表示总的概率越大，完全恢复了原来处理器的可能性就越大。

那么问题来了，clevertick 已经没有足够的时间去恢复 CTEX，所以他将这个任务交给了你这位神犇，希望你能帮他完成。

【输入格式】

该题为提交答案型试题，所有输入数据 *ctex1.in~ctex10.in* 已在试题目录下。

输入的第一行包含四个整数 N, M, S, Z ，分别表示结点数、转移数、文本序列中元素（整数）的范围以及 CTEX 曾经处理过的文本数量。

接下来一行有 N 个整数，第 i 个整数表示编号为 $i - 1$ 的结点的信息，若值为 0，表示它既不是初始结点也不是终止结点，若值为 1 表示它是初始结点，若值为 2 表示它是终止结点。

接下来 M 行，每行两个整数 u 和 v ($0 \leq u, v < N$)，其中第 i 行表示从结点 u 到结点 v 存在一个编号为 $i - 1$ 的转移。

接下来 N 行，每行 $S + 1$ 个非负整数，其中第 i 行的第 j 个整数表示结点 $i - 1$ 上存在类型为 $j - 1$ 的输出装置的概率权值，第 i 行的第 $S + 1$ 个整数表示结点 $i - 1$ 上不存在任何输出装置的概率权值。

接下来 M 行，每行 S 个非负整数，其中第 i 行的第 j 个整数 ($1 \leq i \leq M, 1 \leq j \leq S$) 表示编号为 $i - 1$ 的转移上写有数值 $j - 1$ 的概率权值；

接下来 $2Z$ 行，每两行为一个部分，每部分的第一行表示 CTEX 处理过的一条输入文本，第二行表示对应的输出文本或一个字符串“-1 Error”（表示处理器非正常结束，输出错误信息）。若一行表示文本，那么该行第一个整数 l 表示文本的长度，接下来会有 l 个整数表示这个序列。

【输出格式】

针对给定的 10 个输入文件 *ctex1.in~ctex10.in*，你需要分别提交你的输出文件 *ctex1.out~ctex10.out*。

输出包含两行，表示你给出的 CTEX 构造。

第一行 N 个由空格分隔的整数，分别表示结点 0 到 $N - 1$ 上的输出装置类型，用 -1 表示不存在输出装置。

第二行 M 个由空格分隔的整数，分别表示转移 0 到 $M - 1$ 上写的数值，要求必须在 0 到 $S - 1$ 范围内，且不允许出现从同一个结点出发的两个转移写有相同的数值。

【样例输入】

```
4 8 3 5
1 0 0 2
0 1
0 2
1 1
1 2
2 1
2 2
1 3
2 3
1 1 1 2
1 2 1 2
2 1 1 2
```

```

1 1 2 1
1 1 1
1 1 1
1 1 1
1 1 1
1 1 1
1 1 1
1 1 1
2 2 2
2 2 2
1 0
1 1
3 0 1 1
3 1 0 0
4 0 0 1 1
4 1 1 0 0
6 0 1 0 0 1 0
6 1 0 1 1 0 1
11 1 0 0 1 1 1 0 0 1 0 1
11 0 1 1 0 0 0 1 1 0 1 0

```

【样例输出】

```

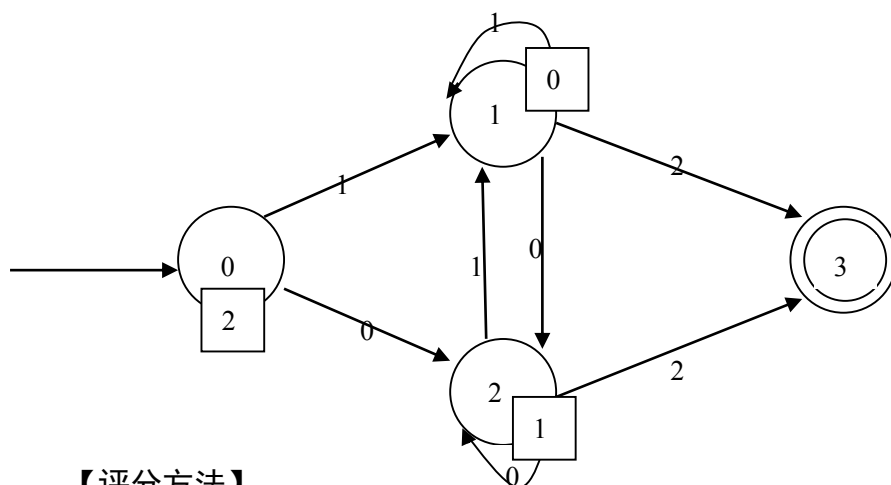
2 0 1 -1
1 0 1 0 1 0 2 2

```

【样例说明】

一种可行方案是构造一个可以将输入文本取反（0 变 1，1 变 0）的处理器（如图）。

该方案概率权值之和为 14，可以证明为满足要求的最小值。

**【评分方法】**

对于每组数据，如果你未提交对应输出文件、提交的输出文件有格式错误，或是按照你给出的输出还原 CTEX 后，其构造不符合题目要求或对输入文件中

给出的任意一组文本不能通过测试（处理器执行的时间超过了 20120526 单位或输出结果与数据中的不符），则得 0 分。

否则，对于每组数据我们都有 9 个参数（参数不会在输入数据中给出），设为 a_1, a_2, \dots, a_9 ，其中 $a_1 > a_2 > \dots > a_9 > 0$ 。设你对于该组数据的输出文件给出的方案中所有的概率权值和为 $total_cost$ ，则该组数据你的得分见下表：

满足的条件	得分
$a_1 < total_cost$	1
$a_2 < total_cost \leq a_1$	2
$a_3 < total_cost \leq a_2$	3
$a_4 < total_cost \leq a_3$	4
$a_5 < total_cost \leq a_4$	5
$a_6 < total_cost \leq a_5$	6
$a_7 < total_cost \leq a_6$	7
$a_8 < total_cost \leq a_7$	8
$a_9 < total_cost \leq a_8$	9
$0 < total_cost \leq a_9$	10

【如何测试你的输出】

我们提供 *checker* 这个工具来测试你的输出文件是否是可接受的。使用这个工具的方法是，首先进入终端，在终端中运行下面的命令进入本题的文件夹

```
cd ctex
```

然后运行：

```
./checker <case_no>
```

其中 *case_no* 是测试数据的编号。

或者运行：

```
./checker <input_file> <output_file>
```

其中 *input_file* 和 *output_file* 分别是输入输出文件的名称。*checker* 将读取名为 *input_file* 的文件作为输入文件、名为 *output_file* 的文件作为输出文件。

之后，*checker* 会将下列信息输出到一个名为 *log* 的文件中：

用输出文件中给出的方案还原 CTEX，然后用还原后的 CTEX 依次处理输入文件中的每个文本，并输出对应结果，每行一个。

每行处理可能的结果为：

1. 处理器正常结束打印的文本；
2. “-1 Error”：处理器非正常结束（注意有的数据要求非正常结束）；
3. “TLE”：其中 TLE 表示处理器执行的步骤超过了 20120526 步，这种情况无论最终是否能够正确打印正确的文本我们都视为错误。

最后会输出总的概率权值之和。

要求给出的输入文件、输出文件格式必须与题目所述一致，且 N, M, S, Z ，每条输入输出文本长度分别不能超过 100000, 300000, 10, 1000 和 100，否则可能输出 “Format error.” 或造成 *checker* 本身段错误。

【特别提示】

请妥善保存输入文件 *ctex*.in* 和你的输出 *ctex*.out*，及时备份，以免误删。