SCRABBLE: Scrable on a Graph

题目描述

我们在图上来玩Scrabble吧！

我们有 K 块字母块，每个字母块上要么有一个大写英文字母，要么为空。空白字母块可以代表任意的英文字母。

我们还有一张 N 个点 M 条边的图。初始时有些点上已经放了字母块，剩下的点是空的，上面没有字母块。每个节点上最多放置一块字母块。

你的目标是将字母块放置在空的节点上，并由字母块构成单词。你可以进行任意轮操作。在每一轮中，你需要放置至少一块字母块，也可以放置多块。放置之后，你需要找到一条路径，要求其**经过至少一个刚刚放置了字母块的节点**，而且路径上每个节点上的字母块顺次拼接得到的单词需要在**提供的词典**中。我们说在这一轮中，你构建了这个单词。同一块空白字母块在不同轮中可以代表不同的字母，但是同样的单词在整局游戏中只能构建一次。

此外，每个节点还有一个得分倍数，在计算得分时，放置在这个节点上的字母块的得分会乘以节点的得分倍数。

每轮中你获得的得分即为你构建的单词的每个字母块的得分（乘以对应得分倍数**后**的得分）之和。单块字母块的得分（乘以得分倍数**前**的得分）根据一般英语写作中字母出现频率确定，详见下文；常见的字母，比如元音字母，只有 1 分，而很少用到的字母，比如Q和Z，则有 10 分。

游戏的总得分即为你每轮的得分之和，你的目标就是最大化你的总得分。为了让游戏更有乐趣，我们将所有字母块随机打乱后排成一排，并称之为剩余字母块队列，每轮中你只能使用剩余字母块队列中的前 20 个字母块（如果不足 20 个，则全部可以使用）。

字母块及其分数

- 0 **分**：所有字母块中 2% 的部分；
- 1 **分**：E × 12%、A × 9%、I × 9%、O × 8%、N × 6%、R × 6%、T × 6%、L × 4%、S × 4%、U × 4%；
- 2 **分**：D × 4%、G × 3%；
- 3 **分**：B × 2%、C × 2%、M × 2%、P × 2%；
- 4 **分**：F × 2%、H × 2%、V × 2%、W × 2%、Y × 2%；
- 5 **分**：K × 1%；
- 8 **分**：J × 1%、X × 1%；
- 10 **分**：Q × 1%、Z × 1%。

为了保证每种字母块的个数均为整数，字母块的总数一定是 100 的倍数。

输入格式

输入数据的第一行包含一个整数 D，代表词典中单词的数量。

接下来 D 行，每行为一个仅包含大写英文字母的字符串，代表一个单词。所有测试数据（样例数据除外）中的词典和 SOWPODS 词典一样，你可以点击链接下载。

接下来一行包含两个整数 N 和 M，由空格分隔，分别代表图中的节点数与边数。

接下来一行包含 N 个整数，由空格分隔，代表每个节点的得分倍数。

接下来一行包含 N 个字符，代表每个节点在初始状态下是否有字母块。有三类可能的字符：

- “.”（点，ASCII 码为 46）：初始时节点上没有字母块；
- “_”（下划线，ASCII 码为 95）：初始时节点上有一个空白字母块；

- 大写英文字母：初始时节点上有一个写着该字母的字母块。

接下来 M 行，每行包含两个整数 X_j 和 Y_j，代表编号为 X_j 和 Y_j 的节点间有一条无向边相连。

最后一行为一个长度不超过 20 的字符串，其中只包含大写英文字母和“_”（下划线，ASCII 码为 95），代表剩余字母块队列中你目前能使用的部分。

输出格式

对于你的每轮操作，按下面的格式输出。在最后一轮结束后，你需要在单独的一行上输出`-1`。

对于每轮操作，首先输出一行，包含一个整数 C，代表你在这一轮中要往 C 个节点上放置字母块。如果你输出的 C 不是正整数，则你的输出会被判为错误答案。

接下来输出 C 行，每行包含一个大写英文字母和一个整数，由空格分割，分别代表你放置的字母块和将其放置到的节点编号。如果你目前无法使用这一字母块，或者你将字母块放置到了非空节点上，则你的输出会被判为错误答案。

接下来输出一行，包含一个整数 L，代表你本轮构建的单词的长度。如果你输出的 L 不是正整数，则你的输出会被判为错误答案。

接下来输出一行，包含 L 个整数，由空格隔开，代表图中一条合法的路径。**这条路径可以不为简单路径**。如果你输出的路径不合法，或者没有经过本轮放置了字母块的节点，则你的输出会被判为错误答案。

接下来输出一行，包含一个长度为 L 的仅包含大写英文字母的字符串，代表你本回合构建的单词。如果这一单词不在词典中，或者已经在之前的某一轮中构建过，或者和你在上面输出的路径不符，则你的输出会被判为错误答案。

如果你的输出满足上述格式，则会被判为通过。

请注意你在不同轮中可以将同一块空白字母块视为不同的字母，以及，你每轮选择的路径必须包含至少一个该轮中放置了字母块的节点。

交互细节

本题为**交互**挑战题，因此，在你输出一轮的操作后，请刷新输出缓冲区。在 C++ 中，你可以通过执行`fflush(stdout);`来刷新缓冲区。

在你输出后，你需要读入一个新的字符串 S，代表你在下一轮中可以使用的字母块。格式同输入格式。

数据范围与约束

测试数据划分成了 4 部分，每部分包含 5 组测试数据。

第 1 部分

- $1 \le N \le 10^4$
- $M = N - 1$
- 两个节点 i 和 j 之间有边直接相连，当且仅当 $|i - j| = 1$
- $K = 10^4$
- 初始时所有节点均为空

第 2 部分

- $1 \le N \le 10^4$
- $M = N - 1$
- 两个节点 i 和 j 之间有边直接相连，当且仅当 $|i - j| = 1$
- $K = 10^4$

第 3 部分

- $1 \le N \le 10^5$
- $M = N - 1$
- 给定的图为一棵树
- $K = 10^5$

第 4 部分

- $1 \le N \le 10^4$
- $N - 1 \le M \le 10^5$
- 给定的图连通
- $K = 10^5$

此外，对于所有测试数据都有：

- $1 \le$ 每个节点的得分倍数 ≤ 10

测试数据的生成

所有数据中的图均为随机生成，其中约 $0 \sim 5\%$ 的节点在初始时放置了字母块。

记剩余字母块队列为 T。初始时 T 包含 K 个字符，每个字符要么是小写英文字母，要么是下划线。每种字母块的数量与“字母块与其分数”中所述一致。生成 T 之后，将 T 随机打乱。

接下来，选取 T 的长度为 0 到 $N \cdot 0.05$ 的前缀，并将这一前缀中的字母块放置到一个随机的节点集合中，不会有两块字母块放置到同一个节点上。之后，将这一前缀从 T 中删去。

每次给定的字符串 S 即为 T 的长度不超过 20 的前缀。当你使用一块字母块时，从 T 中删去其第一个对应的字符。

得分规则

你对于单组测试数据的得分等于该组数据中你每轮的得分之和。

你的总得分为你在所有测试数据中的得分之和。在比赛进行时，你只会被告知前 4 个测试点的得分。

样例数据 #1

初始输入

```
2
ABCDEF
GHIJKL
15 14
```

```
9 8 7 6 5 4 3 2 1 10 1 2 3 4 5
...............
1 2
2 3
3 4
4 5
5 6
6 7
7 8
8 9
9 10
10 11
11 12
12 13
13 14
14 15
ABCDEFGHIJKLMNOPQRST
```

第一轮输出

```
16
1 G
2 H
3 I
4 J
5 K
6 L
6
1 2 3 4 5 6
GHIJKL
```

这一输出代表你构建了单词GHIJKL。然后，在你刷新输出缓冲区后，你需要读入一个新字符串 S：

```
ABCDEFMNOPQRSTUVWXYZ
```

剩余输出

```
6
7 A
8 B
9 C
10 D
11 E
12 F
6
7 8 9 10 11 12
```

```
ABCDEF
-1
```

样例解释

下面说明你构建的两个单词的得分：

- 单词GHIJKL：
  - 字母块 G 放置在 1 号节点，因此其得分为 $9 \times 2 = 18$；
  - 字母块 H 放置在 2 号节点，因此其得分为 $8 \times 4 = 32$；
  - 字母块 I 放置在 3 号节点，因此其得分为 $7 \times 1 = 7$；
  - 字母块 J 放置在 4 号节点，因此其得分为 $6 \times 8 = 48$；
  - 字母块 K 放置在 5 号节点，因此其得分为 $5 \times 5 = 25$；
  - 字母块 L 放置在 6 号节点，因此其得分为 $4 \times 1 = 4$。

  故这一轮的得分为 $18 + 32 + 7 + 48 + 25 + 4 = 134$。
- 单词ABCDEF：
  - 字母块 A 放置在 7 号节点，因此其得分为 $3 \times 1 = 3$；
  - 字母块 B 放置在 8 号节点，因此其得分为 $2 \times 3 = 6$；
  - 字母块 C 放置在 9 号节点，因此其得分为 $1 \times 3 = 3$；
  - 字母块 D 放置在 10 号节点，因此其得分为 $10 \times 2 = 20$；
  - 字母块 E 放置在 11 号节点，因此其得分为 $1 \times 1 = 1$；
  - 字母块 F 放置在 12 号节点，因此其得分为 $2 \times 4 = 8$。

  故这一轮的得分为 $3 + 6 + 3 + 20 + 1 + 8 = 41$。

整局游戏的得分为 $134 + 41 = 175$，即为你在这组数据上的得分。

样例数据 #2

初始输入

```
2
SING
SINGINGS
5 5
1 2 3 4 5
.....
1 2
2 3
3 4
4 5
4 2
GGIINNSS
```

第一轮输出

```
4
1 S
```

```
2 I
3 N
4 G
4
1 2 3 4
SING
```

这一输出代表你构建了单词SING。然后，在你刷新输出缓冲区后，你需要读入一个新字符串 S：

```
GINS
```

剩余输出

```
1
5 S
1 2 3 4 2 3 4 5
SINGINGS
-1
```

输出的路径可以不为简单路径。

时间限制

10 秒

Problem Setter: Sergey Kulik
Problem Tester: Istvan Nagy
Translated by: Hu Zecong