Word ladders
Problem ID: 2ladder

Introduction

Finding the shortest path between two places is a quite common task. It might be the solution to finding a GPS-route or a tool to see how related different subjects are[1]. Therefore efficient algorithms for this task is of course quite crucial.

Aims

The goals of the lab are:

- Implementing BFS.
- Debugging your code.
- Structuring your code in a logical fashion.

Problem formulation

We construct a graph where each node represents a five-letter word (the words are not necessarily in the English dictionary, but consist only of lowercase English letters, a–z). Furthermore we draw an (directed) arc from u to v if all of the last four letters in u are present in v (if there is more than one of a specific letter among the last four letters in u, then at least the same number has to be present in v in order for us to draw the edge). For example there is an edge from ”hello” to ”lolem” but not the other way around. There is both an edge from ”there” to ”where” and the other way around. There is not an edge from ”there” to ”heros” since ”e” is only present once in ”heros”.

You will be asked to answer a series of queries. For each query you will be given two words, the ”starting”-word and the ”ending”-word. The task is to find the length of the shortest path from the ”starting” to the ”ending” word for each query.

Input

The first row of the input consists two integers N, Q, $1 \leq N \leq 10^4$ and $1 \leq Q \leq 500$, the number of words we consider and the number of queries. Then follows N lines containing one five-letter word each. After this Q lines follow containing two space-separated five-letter words each. For each of these lines answer the query.

Output

For each query output a single line with the answer. If there exists a path from the ”starting” to the ”ending” word, print the length of the shortest path. Otherwise print ”Impossible”. Note that you have to write exactly ”Impossible” to get the verdict ”Correct” of the checker.

Sample Input 1

```
5 3
there
where
input
putin
hello
there where
putin input
hello putin
```

Sample Output 1

```
1
1
Impossible
```

[1] https://en.wikipedia.org/wiki/Six_degrees_of_separation