

Bread First Search

Problem ID: breadthfirstsearch

All the Mushroom Kingdom's local N bakeries ran out of bread so Captain Toad went on a quest to buy as much bread as possible. All bakeries form a connected graph with each other, so he decided to apply his favorite algorithm to find the bread: Bread First Search. The algorithm works like a breadth first search for the most part.

Captain Toad takes his trusty pencil and decides to jot down the bakeries that he wants to visit on a grocery list, starting off with the closest bakery 1. He repeatedly visits the bakery at the beginning of his list (and removes the bakery from the beginning of the list when he visits it). When visiting a bakery, he considers all neighboring (adjacent) bakeries in some arbitrary order and adds them to the end of his list (if an adjacent bakery was already in the list, Captain Toad still adds it to the list again), with one small caveat. If a bakery that he is adding has bread, he adds it to the beginning of the list instead, since he wants to visit it earlier. He repeats this process, visiting the bakery at the beginning of his list, until he has exhaustively visited every bakery. If he has already visited the bakery at the top of the list, he skips over it, since there is no point in visiting the same bakery multiple times. Given the graph of bakeries and the order in which Captain Toad went to each bakery, determine the minimum number of bakeries that could have had bread.

For example, in the sample input, Bakery 4 must have contained bread in order for the bakeries to have been visited in that order. Note that while all 4 bakeries could have had bread to satisfy the above ordering, 1 is the minimum amount.

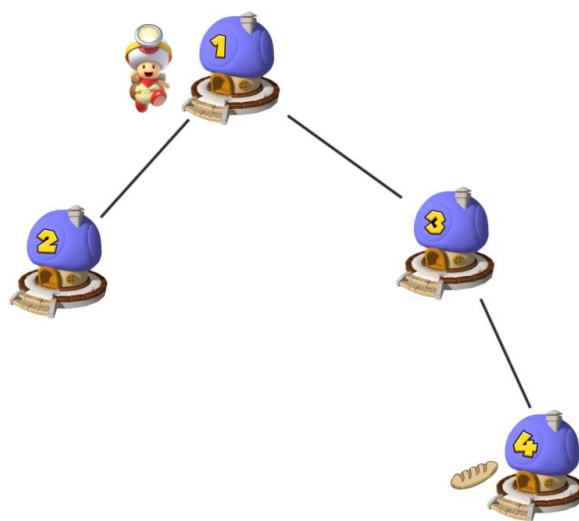
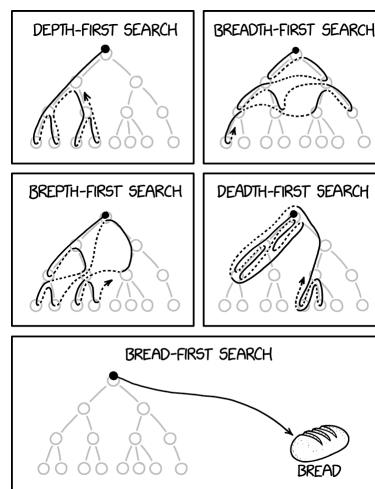


Figure 1: Sample Case 1 Illustration

Input

The first line of the input contains two space-separated integers N ($1 \leq N \leq 2000$) and M ($N - 1 \leq M \leq \min(\frac{N(N-1)}{2}, 5000)$), representing the number of bakeries, and the number of edges connecting the bakeries.

Then M lines follow containing two space-separated integers $v_1 v_2$ ($1 \leq v_1, v_2 \leq N$ and $v_1 \neq v_2$) representing a bidirectional edge between bakery v_1 and v_2 . There are no self-loops or duplicate edges.

The last line of input contains N space-separated integers a_1, a_2, \dots, a_n ($1 \leq a_i \leq N$) representing the order that

the bakeries were visited in. It is guaranteed that all a_i are distinct integers and that the given ordering is a possible traversal of the bakeries under the above conditions for some assignment of bread. It is also guaranteed that $a_1 = 1$.

Output

Output one line containing a single integer M , the minimum number of bakeries that could have had bread given the above ordering.

Sample Input 1

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

Sample Output 1

```
1
```