Problem E
Androids' Frontline

You are the newly appointed commander of the K&G private military contractor group and your first missions is to lead a team to infiltrate the enemy headquarters!

Luckily, you will have some help from allies already positioned in the field, and bases to increase your supplies. The area is split into nodes labelled 0 to $N - 1$. The team you're to lead starts at node 0 and must make their way to node $N - 1$.

Unfortunately, you remember that the team isn't full of robots that can keep on going forever, so you'll need to find the path that uses the least number of movement points so they'll still have the energy to fight once they've arrived. Movement from one node to a connected node costs exactly one movement point. Your team will start with 2 movement points, and for each base on the map, you'll get an additional point.

There's also one more thing to keep in mind: if the team and an ally are on adjacent nodes, they can swap positions without expending any movement points! Knowing this, what is the fewest number of movement points that the team will need to use to reach the enemy base?

Input

The first line contains four integers $1 \leq N \leq 100$, the number of nodes of the map, $0 \leq M \leq \frac{N(N-1)}{2}$, the number of paths of the map, and $0 \leq B < N$, the number of bases on the map, and $0 \leq A < N$, the number of allies on the map.

The next line contains A space separated integers $(1 \leq a_i \leq B)$ where each a_i denotes an ally is currently positioned at node a_i. Nodes 0 and $N - 1$ are guaranteed to never have allies on them.

Each of the next M lines contains two integers $0 \leq u, v < N$, indicating that there is a path from node u to node v.

It is guaranteed that any node will be reachable from any other node, given infinite movement points.

Output

Output a single integer indicating the minimal number of moves required to travel from node 0 to node $N - 1$. If it is impossible to reach the $N - 1$-th node with your current number of movement points, output -1.

```
7 7 0 3
1 2 3
0 1
1 2
2 3
0 4
4 5
3 6
5 6
```

```
1
```