Snap Tokens

Type	Input file	Output file	Time limit	Memory limit
Batch	stdin	stdout	1 second	128 MB

Statement

We play an game of *generalised N-Snap-Tag*[1] using tokens on an **undirected connected** graph of N vertices (numbered 1 to N) and N edges. No edge goes from a vertex to itself and there is at most 1 edge between any two vertices. Initially, there are K identical tokens, each on a different vertex. At each moment, you may choose a token and move it to an adjacent vertex. If two tokens are on the same vertex, they annihilate each other and are **both instantly removed** from the graph. It takes 1 second to move any token.

Given the initial state of a game, determine the minimum amount of time required to remove all tokens from the graph. It is guaranteed that it is always possible to do so.

Input

The first line contains 2 integers N K. The next line contains N integers v_1 ... v_N, indicating for each i that there is an edge between vertices i and v_i. The next line contains K distinct integers p_1 ... p_K, the starting vertices of each token.

Output

Output 1 integer, the minimum amount of time in seconds to remove all tokens.

Sample Input

```
8 4
3 3 4 5 3 5 6 6
1 4 8 7
```

Sample Output

```
4
```

Explaination

It is optimal to move the tokens on vertices 7,8 to vertex 8 and the tokens on 1,4 to vertex 4.

[1] See AIO 2015, *Snap Dragons II*

Constraints

- $2 \leq K \leq N \leq 10^5$
- $1 \leq v_i, p_i \leq N$
- K is even

Subtasks

Number	Points	Max N	Max K	Additional constraints
1	16	20	20	None
2	20	1000	1000	$v_i = (i \bmod N) + 1$ for all i. That is, the graph is a cycle.
3	16	1000	1000	$K = N$. That is, there is a token on every vertex.
4	29	1000	1000	None
5	19	10^5	10^5	None