NAOI TST - Day 1
Problem 4 : SuperPlant DNA

As you are probably aware, human DNA can be represented as a long string over an alphabet of size four (A, C, G, T), where each symbol represents a distinct nucleobase (respectively; adenine, cytosine, guanine, and thymine).

For SuperPlants; colorful plants living in Kouba, however, things are a bit different; research conducted on the latest SuperPlant captured by NASA (National Agricultural and Superplants Administration) revealed that SuperPlant DNA consists of a whopping K distinct nucleobases! SuperPlant DNA may thus be represented as a string over an alphabet of size K.

Now, a certain research group interested in exploiting SuperPlant DNA in artificial intelligence applications has requested to get a single consecutive piece of a SuperPlant DNA string. For R of the nucleobases, they have specified a minimum quantity of how many they need of that particular nucleobase to be present in their sample.

You are interested in finding the shortest substring of the DNA which satisfies their requirements.

Input Specification

The first line contains three integers N, K, and R, denoting the total length of the SuperPlant DNA, the alphabet size, and the number of nucleobases for which the researchers have a minimum quantity requirement, respectively. They obey $1 \leq R \leq K \leq N$.

The second line contains N space-separated integers: the complete SuperPlant DNA string. The i-th of these integers, D_i, indicates what nucleobase is in the i-th position of the DNA string. Nucleobases are 0-indexed, i.e. $0 \leq D_i < K$. Each nucleobase will occur at least once in the DNA string.

Each of the following R lines contains two integers B and Q representing a nucleobase and its mimimum required quantity, respectively ($0 \leq B < K, 1 \leq Q \leq N$). No nucleobase will be listed more than once in these R lines.

Output Specification

Output a single integer, the length of the shortest consecutive substring of the DNA that satisfies the researchers' requirements. If no such substring exists, output "`impossible`".

Constraints

Group	Points	Constraints
1	16	$1 \leq N \leq 100, R \leq 10$
2	24	$1 \leq N \leq 4\,000, R \leq 10$
3	28	$1 \leq N \leq 200\,000, R \leq 10$
4	32	$1 \leq N \leq 200\,000$

Sample Input 1

```
5 2 2
0 1 1 0 1
0 1
1 1
```

Sample Output 1

```
2
```

Sample Input 2

```
13 4 3
1 1 3 2 0 1 2 0 0 0 0 3 1
0 2
2 1
1 2
```

Sample Output 2

```
7
```

Sample Input 3

```
5 3 1
1 2 0 1 2
0 2
```

Sample Output 3

```
impossible
```

Samples Explanation

- In the first sample, there are three substrings of length 2 that contain one each of nucleobases 0 and 1 (namely "`0 1`", "`1 0`" and "`0 1`"), but no substrings of length 1. Thus the shortest length is 2.
- In the second sample, the (unique) optimal substring is "`1 3 2 0 1 2 0`".
- In the third sample, there aren't enough nucleobases of type 0.