Problem S. Check DFS

Time Limit	1000 ms
Mem Limit	131072 kB
Code Length Limit	10240 B

One of the most popular algorithms for traversing a graph is the .
The DFS is usually implemented as a recursive function. First we call the function for the starting node. For each call, we go through the adjacency list of the current node, and recursively call the function for the unvisited neighbours.
Notice that if we permute the adjacency lists of the nodes, the order in which we visit them might differ.
In this problem you are given a graph with N nodes and M edges and a permutation P of size N. If you are allowed to shuffle the adjacency lists, is it possible to visit the nodes during a DFS starting in node 1 in the order given by P?

Standard input

The first line contains two integers N and M.
The second line contains a permutation of size N. The first elements is always equal to 1.
Each of the next M lines contains two integer a and b representing two nodes that share an edge.

Standard output

If it's possible to visit the nodes in the given order print 1, otherwise print 0.

Constraints and notes

- $1 \leq N, M \leq 10^5$
- The graph is connected, simple and undirected

Example 1

Input	Output

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

Example 2

Input	Output

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

Example 3

Input	Output

4 3
1 2 4 3
1 2
2 3
3 4
	0

Example 4

Input	Output

4 6
1 4 2 3
1 2
1 3
1 4
3 2
4 2
3 4
	1