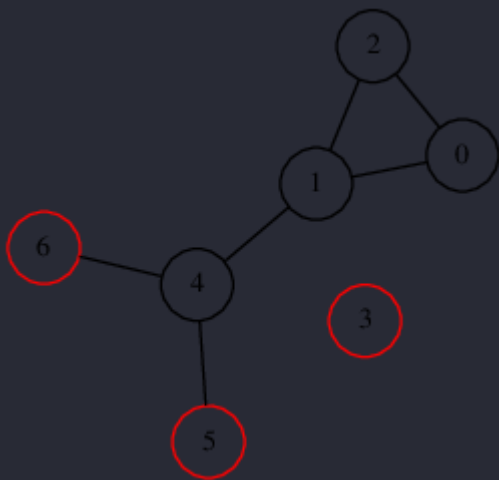


You are given an undirected graph consisting of N vertices, numbered from 0 to $N-1$, connected with M edges. The graph is described by two arrays, A and B , both of length M . A pair $(A[K], B[K])$, for K from 0 to $M-1$, describes an edge between vertex $A[K]$ and vertex $B[K]$.

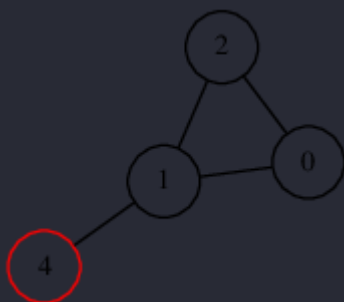
Each second, every vertex with at most one edge connected to it disappears. Every edge which is connected to one of the disappearing vertices also disappears.

After how many seconds will the vertices stop disappearing?

Consider a graph with $N = 7$ vertices and following 6 edges: $(0, 1)$, $(1, 2)$, $(2, 0)$, $(1, 4)$, $(4, 5)$ and $(4, 6)$.



After the first second, vertices 3, 5, and 6 (marked red in the picture above) will disappear:



After the next second vertex 4 will disappear and only vertices 0, 1 and 2 will be left:



All three remaining vertices are connected to two edges, so none of will ever disappear and the answer is 2.

Write a function:

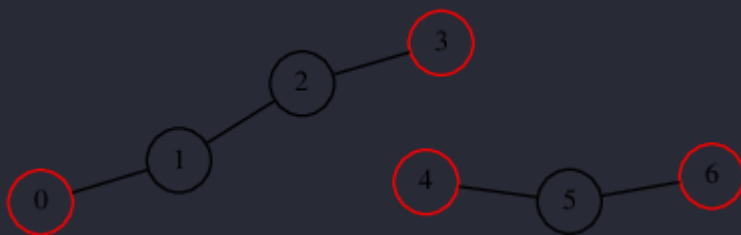
```
def solution(N, A, B)
```

that, given an integer N and two arrays A and B of M integers each, returns the number of seconds after which the vertices will stop disappearing, or 0 if no vertices will ever disappear.

Examples:

1. Given $N = 7$, $A = [0, 1, 2, 1, 4, 4]$ and $B = [1, 2, 0, 4, 5, 6]$, the function should return 2, as explained above.

2. Given $N = 7$, $A = [0, 1, 2, 4, 5]$ and $B = [1, 2, 3, 5, 6]$, the function should return 2. The graph from this example is shown below:

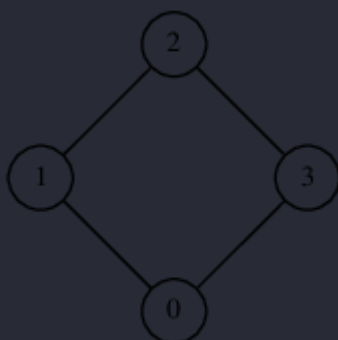


After the first second, vertices 0, 3, 4, and 6 (marked red in the picture above) will disappear:



During the next second, all of the remaining vertices disappear, so the answer is 2.

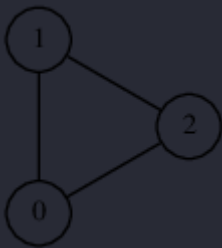
3. Given $N = 4$, $A = [0, 1, 2, 3]$ and $B = [1, 2, 3, 0]$, the function should return 0. Each vertex is connected with two edges, so none of them will disappear.



4. Given $N = 4$, $A = [0, 1, 2]$ and $B = [1, 2, 0]$, the function should return 1.



After the first second, vertex 3 (marked red in the picture above) will disappear. Other vertices are connected with two edges and will never disappear, so the answer is 1.



Write an efficient algorithm for the following assumptions:

- N is an integer within the range $[2..100,000]$;
- M is an integer within the range $[1..100,000]$;
- all elements of arrays A and B are integers within the range $[0..N-1]$;
- there are no self-loops (edges with $A[K] = B[K]$) in the graph;
- there are no multiple edges between the same vertices.