SeaMoney Network Security

Max. score: 100

This problem is no longer available for practice. Apology for any inconvenience!

In SeaMoney Nodes Network, each node ***i*** can directly connect to another node *j* only when *graph[i][j] = 1* .

Some nodes were initially infected with malware. As long as two nodes are directly connected and at least one of them is infected with malware, both nodes will be infected with malware. The spread of this malware will continue until no more nodes can be infected in this manner.

Let ***M***(initial) be the final number of malware-infected nodes in the entire network after the malware stops spreading.

If removing a node from the initial list can minimise ***M***(initial) , return that node. If more than one node satisfies the condition, the node with the smallest index is returned.

Note that if a node has been removed from the list of infected nodes initially, it may still be infected later by malware spread.

Constraints:

1 < graph.length = graph[0].length <= 300

0 <= graph[i][j] == graph[j][i] <= 1

graph[i][i] == 1

1 <= initial.length < graph.length

0 <= initial[i] < graph.length

SAMPLE INPUT

```
3
1 1 0
1 1 0
0 0 1
0 1
```

SAMPLE OUTPUT

```
0
```

Explanation

NA

Time Limit:	1.0 sec(s) for each input file.
Memory Limit:	256 MB
Source Limit:	1024 KB
Marking Scheme:	Score is assigned when all the testcases pass.
Allowed Languages:	Bash, C, C++, C++14, C++17, Clojure, C#, D, Erlang, F#, Go, Groovy, Haskell, Java, Java 8, Java 14, JavaScript(Rhino), JavaScript(Node.js), Julia, Kotlin, Lisp, Lisp (SBCL), Lua, Objective-C, OCaml, Octave, Pascal, Perl, PHP, Python, Python 3, Python 3.8, Racket, Ruby, Rust, Scala, Swift-4.1, Swift, TypeScript, Visual Basic