



Easy

Codewriting

300



Once upon a time, in a kingdom far, far away, there lived a king Byteasar I. As a kind and wise ruler, he did everything in his (unlimited) power to make life of his subjects comfortable and pleasant. One cold evening a messenger arrived to the king's castle with the latest news: all kings in the Kingdoms Union started to enforce traffic laws! In order not to lose his membership in the Union, king Byteasar had to do the same in his kingdom. But what would the citizens think of it?

The king decided to start introducing the changes with something more or less simple: change all the roads in the kingdom from two-directional to one-directional. He personally prepared the `roadRegister` of the new roads, and now he needs to make sure that the road system is convenient and there will be no traffic jams, i.e. each city has the same number of incoming and outgoing roads. As the Hand of the King, you're the one who should check it.

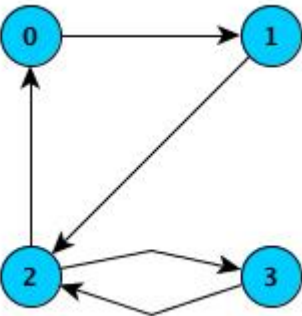
Example

- For

```
roadRegister = [[false, true, false, false],
                [false, false, true, false],
                [true, false, false, true ],
                [false, false, true, false]]
```

the output should be
`newRoadSystem(roadRegister) = true` .

The cities will be connected as follows:



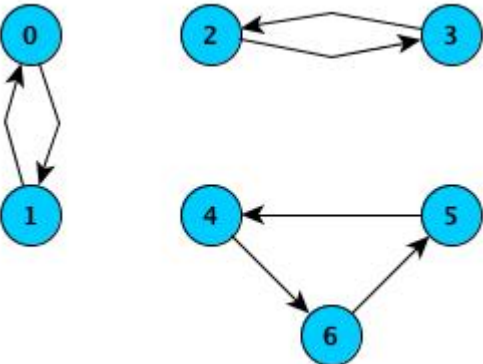
Cities `0` , `1` and `3` (0-based) have one incoming and one outgoing roads, and city `2` has two incoming and two outgoing roads. Thus, the output should be `true` .

- For

```
roadRegister = [[false, true, false, false, false, false, false],
                [true, false, false, false, false, false, false],
                [false, false, false, true, false, false, false],
                [false, false, true, false, false, false, false],
                [false, false, false, false, false, false, true ],
                [false, false, false, false, true, false, false],
                [false, false, false, false, false, true, false]]
```

the output should be
`newRoadSystem(roadRegister) = true` .

The cities will be connected as follows:



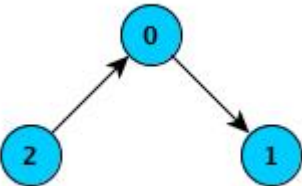
Each city has one incoming and one outgoing road.

- For

```
roadRegister = [[false, true, false],
                [false, false, false],
                [true, false, false]]
```

the output should be
`newRoadSystem(roadRegister) = false` .

The cities will be connected as follows:



City `1` has one incoming and none outgoing roads, and city `2` has one outgoing and no incoming roads.

Input/Output

- [execution time limit] 4 seconds (js)
- [input] `array.array.boolean roadRegister`

Since the cartography is not yet invented in the kingdom, the registers are used instead. A register is stored as a square matrix, with its size equal to the number of cities in the kingdom. If `roadRegister[i][j] = true` , then there is a road from the `ith` to the `jth` city; the road doesn't exist otherwise.

It is guaranteed that there are no loop roads, i.e. roads that lead back to the same city it originated from.

Guaranteed constraints:

`1 ≤ roadRegister.length ≤ 100` ,