Statistics Problem Statement

Problem Statement for SuccessfulMerger

Problem Statement

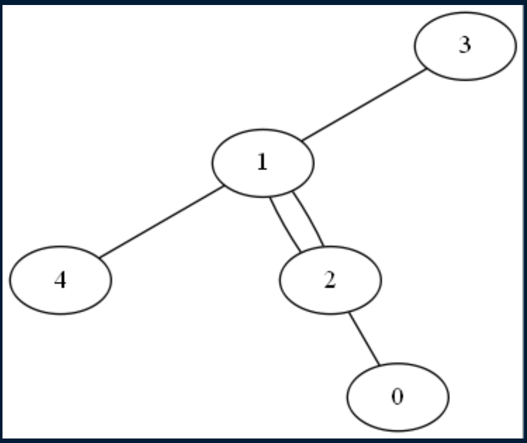
In the country of Byteland, there are N cities connected by N bidirectional roads (that is, the number of roads is the same as the number of cities). The cities are numbered from 0 to N-1. For each i from 0 to N-1, a bidirectional road connects city i and city **road**[i]. (Note that sometimes a pair of cities can be connected by two distinct roads.) It is possible to travel between any pair of cities using the network of roads.

You are the prime minister of Byteland and you must choose one of the cities to be the capital city. The capital city must have the following property: whenever A and B are two distinct cities, it must be impossible to travel between A and B without visiting the capital.

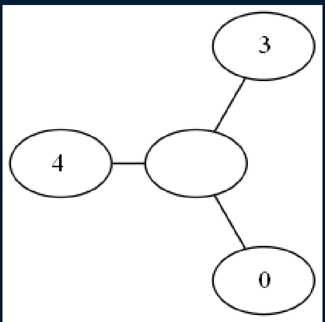
If there is no valid location for the capital, you must create one. In order to do that, you may modify the road network by doing a sequence of mergers. A merger is an operation that merges two adjacent cities into a single new city. Formally, a merger is performed as follows:

- You select two cities X and Y that are directly connected by a road. (They may even be connected by more than one direct road.)
- The two cities are merged together to create a new city Z.
- All direct roads between X and Y disappear. Each other road that connected X or Y to some other city now connects Z to that other city.

For example, suppose this is the network:



There is no valid location for the capital. However, by merging city 1 and city 2, we can create a valid location:



Now the newly-created city is a valid location for the capital.

You are given the description of the road network in the int[] **road**. Please find and return the minimum number of mergers necessary to create a valid location for the capital.

Definition

```
Class:           SuccessfulMerger
Method:          minimumMergers
Parameters:      int[]
Returns:         int
Method signature:int minimumMergers(int[] road)
(be sure your method is public)
```

Constraints

- N will be between 2 and 50, inclusive.
- **road** will contain N elements.
- Each element of **road** will be between 0 and N-1, inclusive.
- For each valid i, **road**[i] does not equal i.
- It will be possible to travel from any city to any other city using the roads.

Examples

```
0)
    {2, 2, 1, 1, 1}
    Returns: 1
    This test case corresponds to the example from the problem statement. In the given road network there is no valid location for the
    capital, so at least one merger is needed. The one merger shown above is therefore the optimal solution for this test case.
1)
    {3, 4, 5, 4, 5, 3}
    Returns: 2
2)
    {1, 0, 1, 0, 0, 0, 1, 0, 1, 1}
    Returns: 1
3)
    {1, 2, 3, 0}
    Returns: 2
4)
    {31, 19, 0, 15, 30, 32, 15, 24, 0, 20, 40, 1, 22, 21, 39, 28, 0, 23, 15, 5, 19, 22, 6, 32, 8, 38, 35, 30, 4, 28, 32, 18, 18, 9, 22, 41, 20, 18, 6, 25, 41, 34, 4}
    Returns: 25
5)
    {1, 0}
    Returns: 0
```

This problem was used for:

Single Round Match 682 Round 1 - Division I, Level Two