Advanced Programming

Samir Datta*

March 31, 2018

Instructions

---

1. Modify the attached file assignment.4.py as *yourusername.4.py*. For example, I would have used aalok.4.py.
2. The file that you submit should be a modification of the attached assignment.4.py file, i.e, it should contain the classes and methods that have already been defined in this file. You may want to define more, but the given ones must be there.
3. The classes and methods provided will be treated as explained in the problem statements and the comments. You are free to change the names for the arguments, but the code will be tested by running the methods as specified in the comments and the code.
4. The *return* type of each function/method should be as specified in the assignment.4.py file.

Problem 1

Implement `countIslands` which counts islands from a given two dimensional string (i.e, string separated by newlines) as described follows:

The 0's represent water, and the 1's represent land. The string is supposed to be thought of as a two-dimensional matrix, where each entry is connected with all the 8 neighbors. An island is a maximally connected land component.

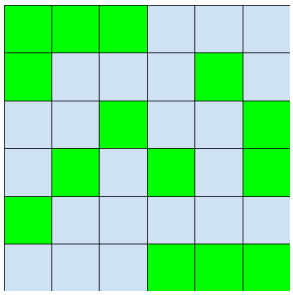
For example, the string below

```
111000
100010
001001
010101
```

*Teaching Assistants - Kishlaya Jaiswal, Agnishom Chattopadhyay

```
100000
000111
```

represents the following grid



and there are 4 islands in it.

Your program may be checked against grids which are 1000 cells high and 1000 cells wide. Your program should terminate in at most 15 seconds.

Problem 2

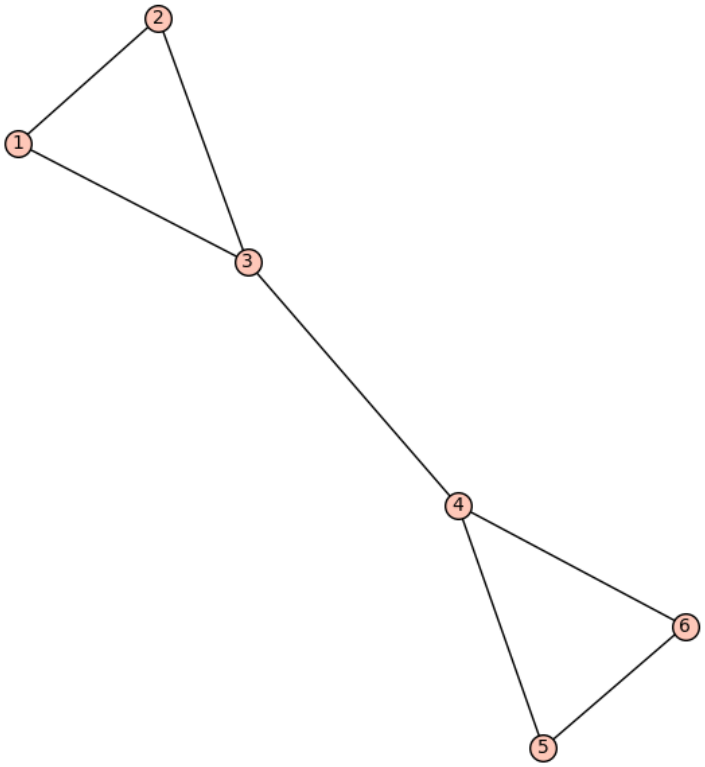
Implement `countBridges` which counts the number of **bridges** in a given graph.

A bridge in a graph is an edge which when deleted increases the number of components.

For example,

`{1: [2, 3], 2: [1, 3], 3: [1, 2, 4], 4: [3, 5, 6], 5: [4, 6], 6: [4, 5] }`

represents



and there is 1 bridge in it.

Your program may be checked against a graph of 100000 vertices. Your program should terminate in at most 15 seconds.

Problem 3

Agnishom is playing a game today (refer to footnote for game's name). Let me introduce you to the rules of the game.

So there is a sequence of stones in front of him, let's call them $(S_1, S_2, \ldots, S_k)$. He is right now standing on the first stone, that is S_1 and he will win the game once he reaches the stone S_k. Each of the stones is marked with a number, say $(N_1, N_2, \ldots, N_k)$. Suppose after some time, he is standing on the stone S_i (which is marked with N_i). Then in the next move, he can either jump to stone S_{i-1} or S_{i+1} or he can directly jump to any stone which is marked with the same number as the one he is standing on (that is N_i). Help Agnishom finish the game as quickly as possible by finding the **minimum** number of moves in which he can reach S_k starting from S_1!

Examples:

- If the marking sequence of stones is $[2018, 1986, 2002, 1847, 2018]$ then the minimum number of moves required is 1 because the following sequence of jumps $(2018, 2018)$ (jump directly from first stone to last stone) is the best possible solution.
- If the marking sequence of stones is $[0, 1, 2, 1, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 3]$ then the minimum number of moves required is 4 because the following sequence of jumps $(0, 1, 1, 3, 3)$ (jump from $1st$ to $2nd$, then $4th$, then $5th$ and finally the $15th$ stone) is the best possible solution.

Hint: This is supposed to be an assignment on graph problems?

---

doki-doki stone club