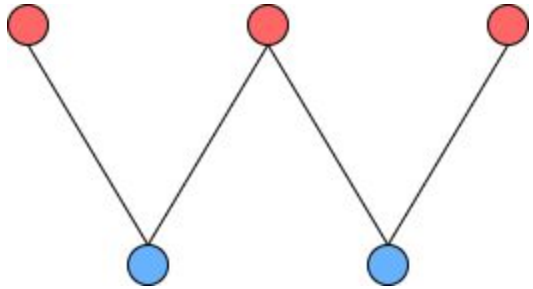
Prime Politics

Problem Explanation

We are given N politicians and M bidirectional edges connecting politicians u and v where 1<=u,v<=N. We have 2 types of currency in cash and have to distribute them between the N politicians in such a way that no 2 adjacent politicians (i.e. politicians who are connected by a bidirectional edge) have the same type of currency. If it is possible to distribute in the above mentioned way, print "SAVED" else print "GONE".

If we connect the politicians with edges to make a graph and keep in mind the above condition, then what we want to find is if a graph is bipartite or not.



Prerequisites

Bipartite Graph

Resources

Refresher on Maximum Bipartite Matching
https://www.cs.cmu.edu/~ckingsf/bioinfo-lectures/matching.pdf

Approach

The question is an application of Bipartite Matching. Use the resource above for a quick refresher on the concept. Keep a boolean variable Flag to denote the possibility of partitioning the Graph and initialise to True.

For each connected component:

1. Run a DFS and assign colors alternately at each recursive call. Since DFS will attempt to visit every node in the graph and backtrack wherever applicable, we can use it with a small modification - we mark the color of a node based on whether we visited it previously or not.

-- If we find a node that is already colored and the color assigned is not the same as the one which we were going to assign, then initialize the Flag to False.

After coloring every node, if we find The Flag set to False, then print “GONE” else print “SAVED” (without quotes).

Time Complexity: O(N)
Space Complexity: O(N)

Where *N* is the total number of politicians