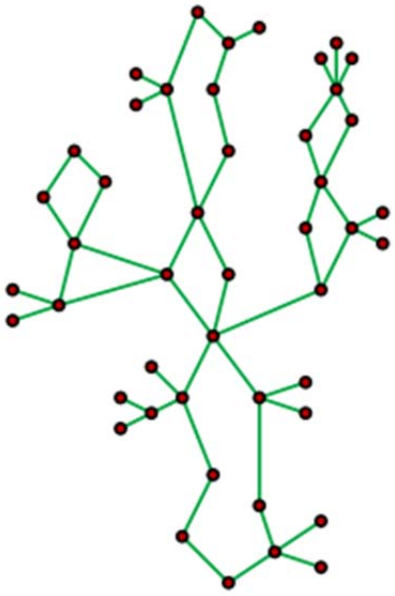
UCF Local Contest — September 3, 2011

A Prickly Problem – Gold Edition

filename: `goldcactus`



A tree is a connected graph in which any two vertices have exactly one path between them.

A cactus (sometimes called a cactus tree) is a connected graph in which any two simple cycles have at most one vertex in common. Equivalently, every edge in such a graph belongs to at most one simple cycle. The graph pictured above is an example of a cactus graph.[1]

A spanning tree can be created from a connected graph by removing a set of edges such that if there are V vertices in the graph, then there are V – 1 edges remaining and every pair of vertices has exactly one path between them. Depending on which edges you choose to remove you will end up with different spanning trees. The cactus graph pictured above contains 36,864 spanning trees.

The Problem:

In this problem, your task is to count the number of spanning trees that a given cactus has. Since this result may be quite large, you should report your result modulo 1,007.

The Input:

Input will begin with an integer T denoting the number of test cases. Each test case will begin with two positive integers $V \leq 100$ and $E \leq (3*V)/2$ denoting the number of vertices and the number of edges, respectively. This will be followed by E lines, each containing an edge in the graph. Each edge is represented by its two vertices and each vertex listed will be between 1 and

[1] Picture of graph and definition of a cactus graph taken from www.wikipedia.org

V (assume that there is at most one edge in the input between any two vertices). It is guaranteed that the graph described in the input will be a cactus.

The Output:

For each test case, output a single line "Case #x: y" where x is the case number starting with 1 and y is the number of spanning trees modulo 1,007. Leave a blank line after the output for each test case. Follow the format illustrated in Sample Output.

Sample Input:

```
3
3 3
1 2
2 3
1 3
5 6
1 2
2 3
1 3
1 4
4 5
1 5
4 3
1 2
1 3
1 4
```

Sample Output:

```
Case #1: 3

Case #2: 9

Case #3: 1
```