

**COP 3503 Recitation #5 Recursion Problems (Week of 6/17)**  
**Due: 6/27/13 (Thursday) at 11:55 PM Webcourses2 time**

**Problem: Latverian Disaster Planning**

A recent infestation of sandworms has endangered the road system of Latveria. Sandworms are nearly indestructible creatures that live underground, and surface every month or so to eat roads. A flock of sandworms will consume an entire road unless it is reinforced using special technology developed by Latveria's Fearless Leader. Recent economic troubles caused by a group of meddling superheroes mean that Latveria lacks the funds to reinforce all their roads. You have been hired to determine which roads are *critical* to the safety and economic well-being of Latveria.

Roads are only built between pairs of cities. A *critical road* is defined as any road that, if destroyed, will make it impossible to travel from some city to another.

By modeling the road network as a graph, we can see that critical roads correspond to edges whose removal disconnects the graph. Recall that a disconnected graph is one with at least two connected components. Your program will figure out which edges of a graph are critical in this sense.

You may use any algorithm and/or data structure you find convenient, as the Glorious Potentate cares only for results.

The program will take file input, from the file 'connectivity.txt'. The file will contain a description of a single graph. For each edge in the graph, your program must output whether the graph is connected if **only** that edge is removed. (In other words, it must solve the problem for every edge in the graph.)

The first line of the input file will contain two integers  $V$  and  $E$ , the number of cities and roads in Latveria. This is followed by  $E$  lines, each containing two integers - the two cities connected by that edge. Latveria doesn't bother with names for cities, and simply numbers them using zero-based indexing. See the sample input for an example.

Your program must write the results to standard output. It must print only those roads that are critical, one road on each line. The relative order should be the same as in the input. See sample output for an example.

**What to turn in**

Turn in the file DisasterPlanning.java. Please follow all the specifications and don't forget to put comments into the code. Using Big-O notation, indicate the time complexity in terms of the appropriate variables.

**Sample Input**

9 10  
0 3  
0 1  
1 2  
2 4  
1 4  
4 6  
4 5  
5 7  
5 8  
8 7

**Sample Output**

0 3  
0 1  
4 6  
4 5