CS202 HW 4

1. Written assignment: Printed textbook 3.8 and 3.15.

2. Coding assignment: Given a map of n cities (labeled $1, \ldots, n$) and m two-way roads between cities. Find out how many new roads are needed so that all cities are connected, that is, it is possible to go from any city to any other city following the roads. If all cities are already connected, output 0.

   The input is given in a text file "map.txt". **The first line in the file is n (the number of vertices) and m (the number of edges)**, followed by m lines, where each line consists of two integers A and B, indicating that there is a road between A and B. It is guaranteed that both A and B are between 1 and n inclusive.

   The output is a single number, which is the minimum number of new roads needed.

   Write a program to solve this problem. A simple sample input and output are given below. However, your program should be able to take any input file that satisfies the above input specifications and print the correct result. One large input file "map.txt" is attached. **But, you should write a program to large random graphs and test your program on them**.

   Note that for this assignment, the DFS algorithm should be implemented iteratively (using loop instead of recursion), because deep recursions will result in the `Stack Overflow` error in Java.

   Analyze the running time of your algorithm.

   Sample input:

   ```
   5 3
   1 2
   3 1
   3 2
   ```

   Sample output:

   ```
   2
   ```

   Explanation: In this map, three cities 1, 2, and 3 form a triangle. Two cities 4 and 5 are isolated. So two new roads are needed so that all cities are connected.