**Coding Project – Graph**

**Due Date:** April 15th, 2019

**Instruction**

Please write your code in Java. When submitting, please submit your original code file, not pasting them to word document or pdf.

**Number of Islands**

Given a 2d grid map of 1s(land) and 0s(water), count the number of islands. An island is surrounded by water and is formed by connecting adjacent lands horizontally or vertically. You may assume all four edges of the grid are all surrounded by water.

Example:

Input:

11000

11000

00100

00011

Output: 3

**Biconnectivity**

Given a graph G, check if the graph is biconnected or not. If it is not, identify all the articulation points. The algorithm should run in linear time.

**Graph Bipartite**

Given an undirected graph, return true if and only if it is bipartite.

The graph is given in the following form: graph[i] is a list of indexes j for which the edge between nodes i and j exists. Each node is an integer between 0 and graph.length – 1. There are no self edges or parallel edges: graph[i] does not contain i, and it doesn’t contain any element twice.

Input: [[1,3], [0,2], [1,3], [0,2]]

Output: true

**Maze**

A maze will be given by a 2d integer matrix(int [][]) that only contains 1s and 0s. 1 indicates a block that you cannot pass and 0 indicates a clear space that you can pass from.

You should implement a program that reads the given matrix and takes any two points as inputs and tells whether there is a path in this maze between such points. Note that index starts at 0 for the points. For example, given the start point (1, 1) and the end point (1, 8), your program outputs ‘YES’ if there exists at least one clear path between them and ‘NO’ if no path exists.