Design and Analysis of Algorithms: Program Assignment 1

CS 3310 **Fall 2025**

Assignment Weight: 4.0

Functional Requirements

Write a Java program (based on a graph traversal algorithm you've learned in this class) that, for a given undirected graph, outputs: the vertices of each connected component.

Your programs should take inputs from a file via the command line with the following structure in the input file. Each line of the input file represents a graph. The first number in a line specifies the number of vertices in the graph. Then pairs of vertices define the edges.

An example of an input file is as follows:

5 (1,2) (3,4) (3,5) (4,5)
4 (1,2) (2,3) (1,4)

It specifies two graphs. The first graph has five vertices (1,2,3,4,5) and four edges. The second graph has four vertices (1,2,3,4) and three edges.

Proper output should look (something) like:

– –
Graph1:
Two connected components: {1 2} {3 4 5}

Graph2:
One connected component: {1 2 3 4}
– –

You must test your programs on a nontrivial input file (with at least 3 graphs and each graph having 7-10 nodes). Your output should be formatted nicely so that it is easy to read.

Your program should use "good style". See the separate handout on style requirements for CS-3310 programs.

Additional Requirements

Additionally, you should create the following ordinary text file:

README: Information on how to compile and run your program.

Submitting Your Program

Before 11:59:59 p.m., Friday, September 26, 2025, you must upload a zip archive to the course Canvas assignment for Programming Assignment 1. This zip archive must contain all source code files for your program, including a class named `Prog1` with a `main` method.

No code printouts are required.

Hints

Connected component: in graph theory, a *connected component* (or just *component*) of an undirected graph is a subgraph in which any two vertices are connected to each other by paths, and which is connected to no additional vertices in the supergraph.