Testing Document

Objectives of Testing

This project was developed using test driven development. Therefore a test for each module was written before coding for that module ever began.

Methods Used

JUnit was used to conduct unit testing. The major components of this project were tested. The following is the breakdown of test files included in the test package of the project, and a description of what they tested:

AListTest: this tested the addition of edges and vertices into the adjacency list as well as deletion.

DFSTest: this allows the user to see the output the DFS through each iteration. This file uses the example in podcast 14E.

DirectedGraphTest: this tested the graph operations of the ADT.

MetricsTest: this tested the output for each run as specified in the assignment. The output for SSC-Test.vna was used as assertions.

Hardware/Software Environment

The project was developed in Eclipse Juno, but the compiler was configured to run with Java 1.5. Testing was done on a laptop with 8 GB of RAM. The following was added to the eclipse.ini file for performance enhancement:

-Xms1536m

-Xmx2048m

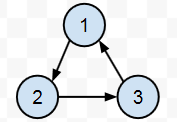
-Xss2m

-XX:MaxPermSize=512m

-XX:+UseParallelGC

Test Case Specification

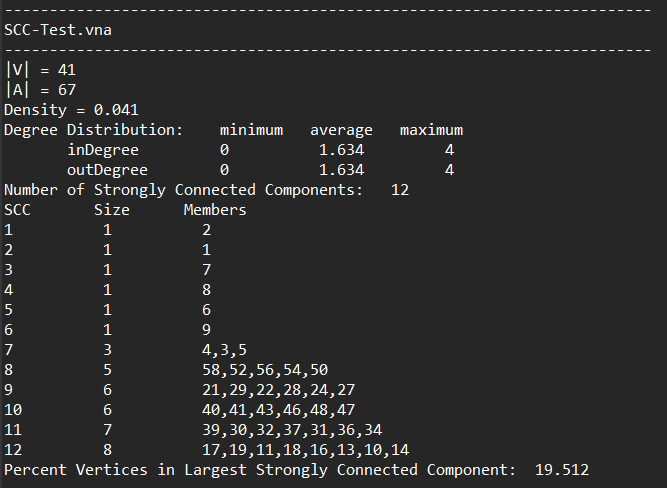
The graphs used for the JUnit tests was the SCC-Test.vna graph (used in MetricsTest.java) and a simple graph as shown below:

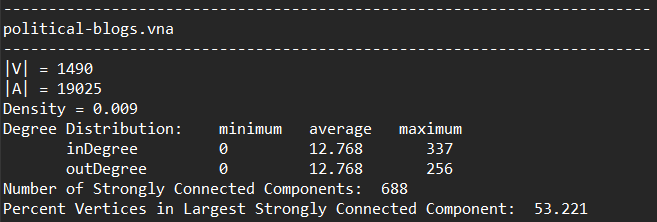


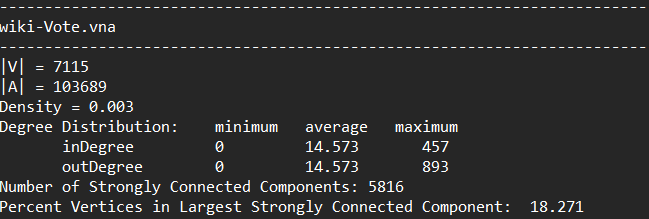
The file test.vna depicts this graph as well. A disconnected graph and a graph with no edges was also used to the test graph ADT (DirectedGraphTest.java). In some cases, an additional fourth vertex was added.

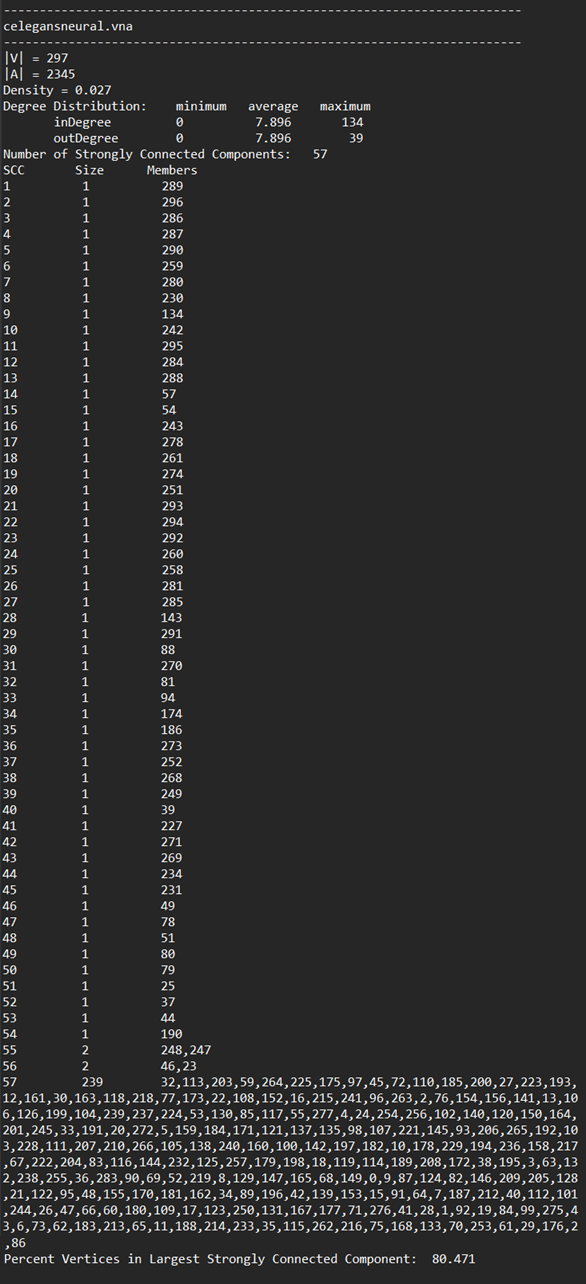
Graph Results

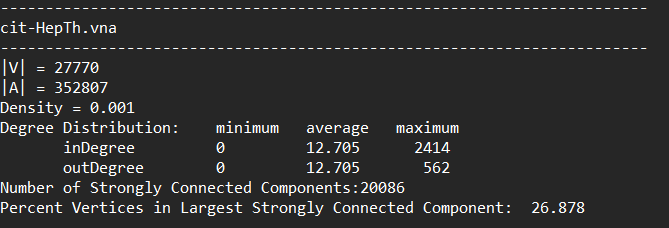
The results for the graphs (up to soc-SlashDot0902.vna) are shown. Files after this file were not tested, as run time became an issue (excess of 30 minutes or more).

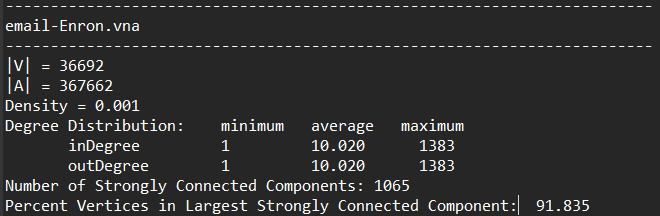


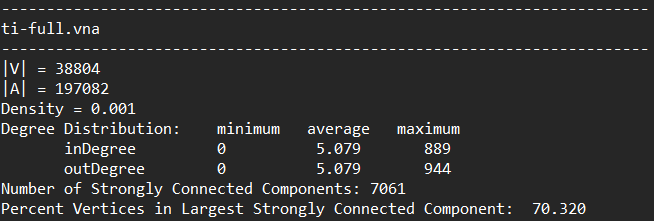


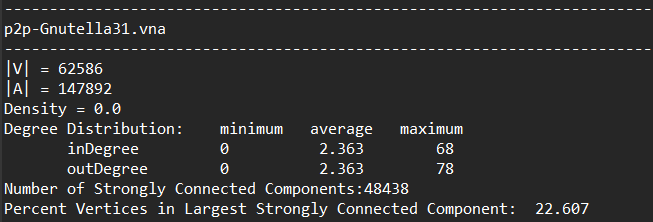


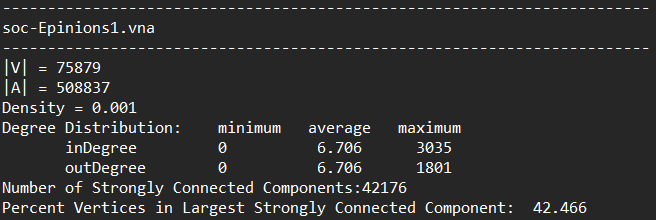












Discussion of the Results