COP 4531: Complexity and Analysis of Data Structures and Algorithms

Programming Assignment 2, Due Date - 10/24/2013 11:59 PM

Assigned - 10/10/2013

In this assignment you will implement two important algorithms related to graphs. We have discussed algorithms for computing the connected components and the strongly connected components of a graph. Note that connected components applies to undirected graphs whereas strongly connected components applies to directed graphs. Recall that given a directed graph a weakly connected component of the graph is the connected component of the underlying undirected graph. Also recall that a graph may have more than one connected component or strongly connected component. In this exercise we will write code to find all the weakly connected components and strongly connected components of a directed graph. Then we will report the following statistics for the largest weakly connected component (WCC) and the largest strongly connected component (SCC) - the number of nodes and edges in the largest WCC and the number of nodes and edges in the largest SCC.

A sample output is as follows:

```
Statistics for given graph is as follows:
Largest WCC
Nodes - 10
Edges - 9
Largest SCC
Nodes - 19
Edges - 18
End
```

Test Cases

We will test your code on the Web Graph of www.stanford.edu that was scraped and put on Blackboard inside CourseLibrary/ProgAssignment 2 Data/. The code should print out the correct values of the statistics we are looking for. Make sure that you test your code on hand crafted examples of directed graphs so that you know that it works correctly on smaller graphs which are easier to test while debugging. We will not supply the correct values of the statistics we are looking for on the Stanford graph. Your code should be tested on smaller graphs and you should be sure that it works correctly on smaller examples, then you will know that it will work correctly on larger ones.

In case you really want to test it on a larger graph, there is another web graph on Blackboard in the same location named Notre Dame Web Graph that is an older web graph of the University of Notre Dame web site. You can test your code on this graph. The statistics are:
Nodes in largest WCC 325729
Edges in largest WCC 1497134
Nodes in largest SCC 53968
Edges in largest SCC 304685
Note that you will need to write parsers for parsing the supplied graph files. This should also be a part of your submission.

Submission Guidelines

You are free to use any programming language for the implementation. Your submission must have the following:

- A one page report discussing the algorithm that you have implemented and the problems that you have faced in trying to do so.
- A README file that describes how the code can be compiled and run and also lists any external dependencies that needs to be satisfied for compiling and running the code.
- A Makefile that can be used for compilation, I will not write Makefiles for compiling your code neither will I write any commands except for single word commands like **make** for compiling your code.
- Your submission should have a single point of entry for running the code, like a main() method in C or its equivalent in any other language.
- The total points for the assignment is 100 which is divided as follows:
  - Report - 20 Points
  - Design of program - 10 Points
  - Error free compilation - 20 Points
  - Correctness on supplied dataset 50 Points
- Please mail in your submission to tm09@my.fsu.edu by 11:59 PM on Thursday October 24, 2013. DO NOT SUBMIT THROUGH DROPBOX. They will not be accepted and will result in late penalty.