CS 494/594/690, Graph Algorithms, Applications and Implementations
Spring 2017, Homework 1

1. Design an algorithm to take as input a graph G of order n, and determine whether it contains K_3 as a subgraph. State your algorithm with reasonable precision in plain English or very high-level pseudocode. Do not program it. Don't fret over efficiency. As a function of n, how long does your method take in the worst case?

Put your algorithm description and answer in a plain text file named README.txt.

2. Write a program that reads a simple, unweighted, undirected graph from a file and outputs to standard output the minimum degree, maximum degree, and density of the graph.

Use of C/C++ is recommended, but you can choose any programming language you wish, with the following stipulations.

1) Your program must compile and run when invoked from the Linux command line on the EECS Linux machines, using only software currently installed. I will test your code on one of the Hydra Lab machines.
2) Your program should take the name of a graph file as a command-line argument.
3) Do not use any library routines specifically designed for graphs (e.g. Boost).
4) Include an example how to invoke your program in your README.txt file.

Programming languages that are confirmed to work on the lab machines include C/C++, Java, Perl and Python. An example graph file is attached. A user should see something very similar to the following when invoking your program (assuming your executable is named "degree").

```
>./degree graph1.txt
Minimum degree: 2
Maximum degree: 5
Density: 0.53333333
>
```

Submit your program by emailing all files necessary to compile and run your code to cphill25@utk.edu prior to the beginning of class next Wednesday, January 18. Include your README.txt file. If you have any questions, please do not hesitate to email me or drop by during office hours.