CS 594/690, Graph Algorithms, Applications and Implementations
Spring 2017, Homework 6

1. The global clustering coefficient is defined as $g = a / (a + b)$, where a is the number of closed P_3's and b is the number of open P_3's.

    a. Draw the densest 6-vertex graph for which $g = 0$.
    b. Draw the sparsest 6-vertex graph for which $g = 1$.
    c. Describe in 15 words or fewer the family of graphs for which $g = 1$.

2. Write a program to compute the Jaccard similarity between two vertices in an undirected, unweighted graph. Recall that Jaccard(u, v) = $|S \cap T| / |S \cup T|$, where S is the set of neighbors of u and T is the set of neighbors of v. (Note that v may or may not be a member of S and u may or may not be a member of T.) Your program should take three command line arguments: the name of a graph file followed by the indices of two vertices. It should output the Jaccard similarity between the two vertices. The output should look very similar to the following two examples.

```
>./jaccard graph6.txt 0 1
0.33333333333
>

>./jaccard graph6.txt 1 3
0.25
>
```

3. Write a program to compute both local and global clustering coefficients in an undirected, unweighted graph. The local clustering coefficient of a vertex v is $2 \cdot N_v / [d(v) \cdot (d(v)-1)]$, where N_v is the number of edges between neighbors of v and $d(v)$ is the degree of v. Your program should take two command line arguments: the name of a graph file followed by the index of a vertex. It should output the local clustering coefficient of the vertex as well as the global clustering coefficient of the graph. The output should look very similar to the following.

```
>./clustercoefficient graph6.txt 2
Local: 0.3
Global: 0.5294117647
>
```

The following apply as usual.

- You may choose any programming language you wish, as long as your program compiles and runs 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.
- Do not use any library routines specifically designed for graphs (e.g. Boost).

- Include an example how to compile and/or invoke your program in a README.txt file.

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, February 8. Bring a hard copy (either printed or handwritten) of your answers to question 1 to class. If you have any questions, please do not hesitate to email me or drop by during office hours.