CS 494/594, Graph Algorithms, Applications and Implementations
Spring 2015, Homework 6

Implement a program that finds a maximal matching in a simple, undirected, unweighted graph. The program should construct a maximal matching in four different ways.

1. Choose a highest degree vertex and its highest degree neighbor.
2. Choose a lowest degree vertex and its lowest degree neighbor.
3. Choose a random edge to place in the matching.
4. A method of your choice. Several possible options were discussed in class. Describe your algorithm unambiguously.

The program should create a maximal matching in one run, using all four methods. The output should look similar to the following.

```
>./matching somegraph.txt
Highest Degree: 4 edges
1-2, 3-5, 0-8, 4-6

Lowest Degree: 5 edges
8-9, 5-6, 3-4, 2-7, 0-1

Random Edges: 4 edges
1-2, 3-5, 0-8, 4-7

My Method: 5 edges
8-9, 5-6, 3-4, 2-7, 0-1
```

Test your codes on these files:

C_14.txt	random_500.005.txt
K_20.txt	random_500.2.txt
Peterson.txt	parkinson_67.03.txt
S_5_K_4.txt	random_100.1.txt
graph3.txt	K_3_3_3_3_3.txt

Prepare a brief report on your findings. Include a table showing the number of edges placed in each graph's maximal matching by each of the four methods. Can you deduce from this any graph characteristics that favor one method over others? Are there any types of graphs on which any of the methods are guaranteed to find a maximum matching?

As usual, your program should take a file name as a command-line argument and output to standard output. All graph files will be in the format discussed in class. All stipulations regarding the first two homework assignments remain in effect. Be sure to test that your programs compile and run in Linux on the EECS lab machines.

Email your source code and any makefile, along with your report in .pdf, .doc. docx or .txt format to cphill25@utk.edu prior to the beginning of class next Wednesday, February 18. If you have any questions, please do not hesitate to email me or drop by during office hours.