CS 494/594, Graph Algorithms, Applications and Implementations
Sprint 2015, Homework 2

Note that what follows is a description of one program, broken into two steps to help guide your implementation. If you are comfortable implementing the entire program in one pass, by all means feel free.

1. Write a program that performs a depth-first search (DFS) on a connected, simple, undirected, unweighted graph. The program should output to standard output all vertices in the graph in the order visited by the DFS. The DFS should
        a. Begin with vertex 0.
        b. Choose the lowest-numbered unvisited neighbor to visit next.

2. Extend your program to perform a DFS on each connected component of the graph. Each DFS should begin with the lowest-numbered unvisited vertex remaining in the graph, and otherwise use the same rules as above. Report the DFS sequence for each component, with a single space between each vertex number, one connected component per line.

Your program should read the graph from a file as per homework 1.

Use of C/C++ is encouraged, 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 any of the EECS Ubuntu machines, using only software currently installed.
2) Your program must take the name of a graph file as a command-line argument.
3) Your program cannot use any data structure or routine from any graph library (e.g. Boost).
4) Your program must read the graph into an adjacency matrix, clearly commented as such in your code.

An example graph file is attached. Users should see something like the following when invoking your program on the example graph file (assuming your executable is named "dfs"). The following output is correct for the example graph file.

```
>./dfs graph2.txt
0 11 3 6 7 8
1 5 12
2 9 10 4
>
```

Submit your program by emailing all necessary source code (including any needed makefile) and any other files necessary to compile and run your code to cphill25@utk.edu prior to the beginning of class next Wednesday, January 21. If you have any questions, please do not hesitate to email me or drop by during office hours.