Algorithms - CS5800 - Summer 2, 2017 - Programming assignment

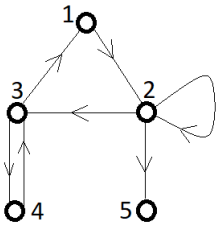
1. **PROGRAMMING ASSIGNMENT**
   The deadline for the programming assignment is Aug 14 2017.
   In this assignment you will write a computer program that gets as input a text file that describes a directed graph (in a manner that will be defined below) and computes the strongly connected components of the graph.

   Input: The input to your program is a text file that describes a directed graph in the following way.

   (a) The first line of the file includes two integers n and m which indicates the number of vertices in the graph and the number of edges in the graph respectively. You may assume that $1 \leq n, m \leq 2^{31} - 1$.
   The integers are separated by one blank character (one space).
   The vertices of the graph will be named $1, 2, ..., n$.

   (b) Starting at the second line, each line contains 2 integers i and j such that $1 \leq i, j \leq n$. The integers are separated by one blank character (one space).
   Each line i j represents an edge $i \rightarrow j$ in the graph.

   (c) For example, the file:
   5 7
   1 2
   2 2
   2 3
   2 5
   3 1
   3 4
   4 3
   corresponds to the following graph:



   (Note that there might be an edge from a vertex to itself.)

   Output: The output of your grogram should be a text file (with the name "components.txt"), that describes the strongly connected components of the input graph G.
   The structure of the output file should be as follows: For each strongly connected component the file will have one line with the text "Component" followed

with lines with the numbers of the vertices in the strongly connected component. Each line will contain one integer with the number of a vertex in the strongly connected component.
For example, the output for the above graph should be the following text file:
Component
1
2
3
4
Component
5

Remarks:

(a) You may use any programming language that can be run on the departments computers.
We suggest that you use one of the following languages:
C, C++, Java, Python.

(b) You should implement all the algorithms that you use. You should not use built in algorithms and/or non-trivial built in data structures.
If you use a programming language that provides built in, high level algorithms, like sorting algorithms, or data structures like heaps, you must not use them.

(c) You may assume that the input file is in the same directory as your program.

(d) IMPORTANT: You should provide a short explanation (in a pdf, doc, or some other similar file) of how your program works and how to run it. You should list the algorithms that you used, and explain their implementation.