CS-UY 2413: Design&Analysis of Algorithms—Spring 2017 Prof. Boris Aronov
Department of Computer Science and Engineering Prof. Lisa Hellerstein

Homework 6

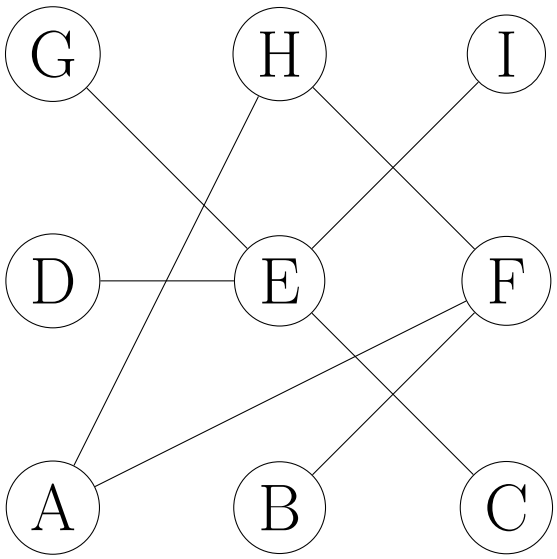
Due April 3 at the start of lecture (Aronov's sections).
Due April 4 at the start of lecture (Hellerstein's section).
This homework should be handed in on paper. No late homeworks accepted. Contact your professor for special circumstances.

Policy on collaboration on this homework: The policy for collaboration on this homework is the same as in previous homeworks. By handing in this homework, you accept that policy. Remember: A maximum of 3 people per group.

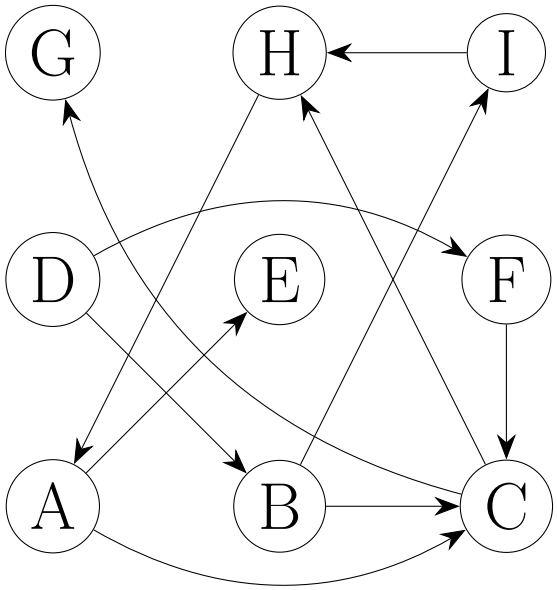
Notes: (i) Every answer has to be justified unless otherwise stated. Show your work! All performance estimates (running times, etc) should be in asymptotic notation unless otherwise noted.

(ii) You may use any theorem/property/fact proven in class or in the textbook. You do not need to re-prove any of them.

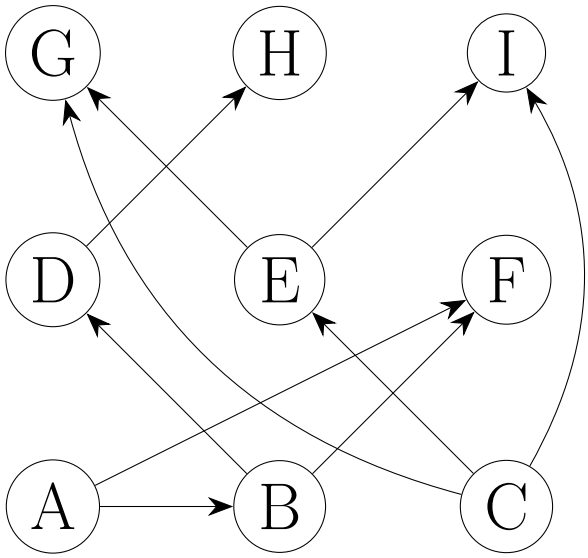
1. Run a DFS on the given *undirected* graph, starting with vertex A. Draw the DFS tree/forest and label every vertex with entry/exit times (pre/post times, in textbook terminology), as shown in class; when there is choice in visiting a vertex's neighbors, visit them in alphabetical order. If the first top-level call to DFS does not visit all the vertices, repeatedly invoke it on the alphabetically next unvisited vertex. Only show the final tree/forest, no need to show the stack contents, or other details of running the algorithm.



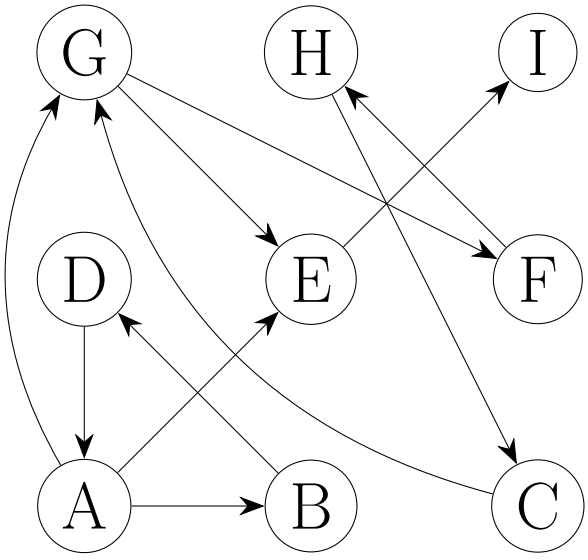
2. Repeat for the given *directed* graph. Draw separately the DFS tree/forest. Label each graph edge as a *tree (T)/forward (F)/backward (B)/cross (X)* edge.



3. Consider the given directed graph. Demonstrate that it is acyclic by constructing a topological sort OR show that it is not by finding a cycle. Use a DFS-based algorithm.



4. On the given directed graph, run the strongly connected components (SCC) algorithm described in the textbook. Show the first DFS forest, the ordering used by the final stage and the resulting strongly connected components.



5. For each of the following situations, state whether it is possible. If not, explain why not. If yes, give a (small) example when it happens:

   (a) In an undirected graph, you remove an edge, and the number of connected components decreases.

   (b) In an undirected graph, you remove an edge, and the number of connected components stays the same.

   (c) In a directed graph, you remove an edge, and the number of strongly connected components stays the same.

   (d) In a directed graph, you remove an edge, and the number of strongly connected components increases by 2 or more.

   (e) In a directed graph, with every vertex reachable from the vertex s, you remove an edge, and the number of vertices reachable from s (other than s itself) goes down to 0.

6. You are working with investigators on a legal case involving text messages. There is a collection of n text messages, numbered 1 through n, all sent at different times. Noone knows when the messages were written. However, for some pairs of messages, the investigators have evidence that one was written before the other. Based on this evidence, they have created a file with information about these pairs. Each line of the file has the form (message $i <$ message j), which means that message number i was written before message number j.

   The investigators want you to use the data in the file to determine the answer to the following question, which we will call Question A.

   Question A: Could all the information in the file be correct? That is, is there a potential ordering of the messages, from earliest to latest, that is consistent with all the information given in the file?

   This question can be answered by using the information in the file to build a graph, and then running an appropriate graph algorithm. The following problems ask you to explain, at a high level, how this could be done. (Your answers to all of the following questions should be short, and should NOT include information about low-level programming issues such as reading in the data file.)

   (a) What are the vertices of the graph and what are the edges of the graph? Explain the relationship between the information in the file and the vertices and edges of the graph. Also specify whether the graph is directed or undirected.

   (b) Which graph algorithm should you run to answer Question A? Just give the name, or a very short description, of the graph algorithm. You do NOT have to give pseudocode.

   Then describe how would you use the output of the graph algorithm to answer Question A.

(c) Suppose all the information in the file is correct. Consider the following new question.

Question B: Given two messages, i and j, could message i have been written before message j? That is, is there a potential ordering of the messages, from earliest to latest, consistent with all the information in the file, where message i is listed somewhere before message j?

Describe how to use a standard graph algorithm to answer Question B. You should NOT give pseudocode for the standard graph algorithm or describe how it works, but you do need to be clear about how you are *using* that algorithm to get the answer to Question B.