

Due: Friday 11:59 am, Feb. 18, 2022

Instructions: You may work in groups of up to three people to solve the homework. You must write your own solutions and explicitly acknowledge everyone whom you have worked with or who has given you any significant ideas about your solutions. You may also use books or online resources to help solve homework problems. All consulted references must be acknowledged. The acknowledgements need to be made by answering Problem 0 below.

You are encouraged to solve the problem sets on your own using only the textbook and lecture notes as a reference. This will give you the best chance of doing well on the exams. Relying too much on the help of group members or on online resources will hinder your performance on the exams.

Submissions being late in 2 hours will be accepted with a 20% penalty. Submissions late more than 2 hours will receive 0. There will be no exceptions to this policy, as we post the solutions soon after the deadline.

For the full policy on assignments, please consult the syllabus.

Formatting: Start a new page for each problem.

Describing an Algorithm: Please make sure you use plain wording to explain your algorithm. It is always a good practice to start with a summary of the high-level idea of your algorithm to ease graders understand your solution quickly. Then, explain your algorithm, using plain wording and including enough details.

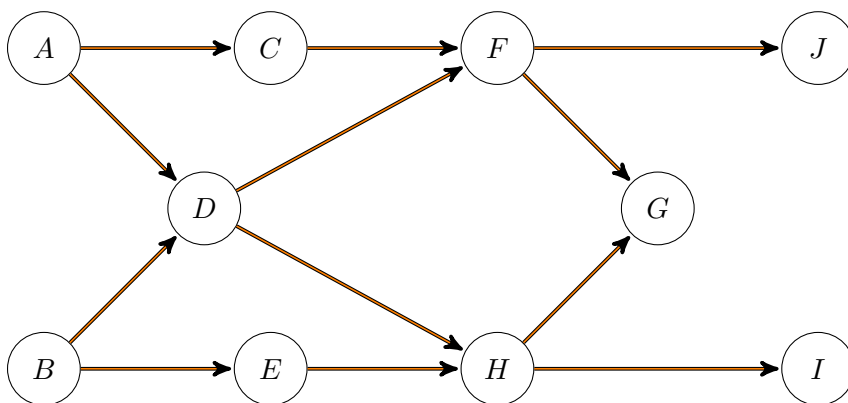
The use of pseudo-code is optional, and it is your decision. No matter you use it or not, above description in words is always required. The pseudo-code has its own advantage in explaining structured (i.e., if-else, for-loop, recursive functions, etc) algorithms and in putting details in the right place. If you think pseudo-code better explains your algorithm, and/or helps graders understand your solution, and/or contains more details not included in the plain-wording description, then use pseudo-code. If you think everything is already clearly explained in the description with words, then you don't need to include pseudo-code. An algorithm that is only written in pseudo-code (i.e., missing above plain-wording description) is not acceptable, as it is extremely hard to read just pseudo-code without any explanation.

Here is a general situation that may help you decide whether to use pseudo-code or not. An algorithm could be "designed from scratch", i.e., you will need to come up with the step-by-step procedure. This usually involves in implementing a function with clear input and output. In this case, including pseudo-code usually helps. All algorithm we've seen so far (e.g., merge-two-sorted-arrays, merge-sort, etc) falls in this category. Second, an algorithm could also be "transformed into another algorithm", i.e., you use an existing algorithm to solve this problem. In this case you usually don't need to include pseudo-code but to describe how to transform one problem into the other. We will see such examples soon.

0. (0 pts.) Acknowledgements. The assignment will receive a 0 if this question is not answered.

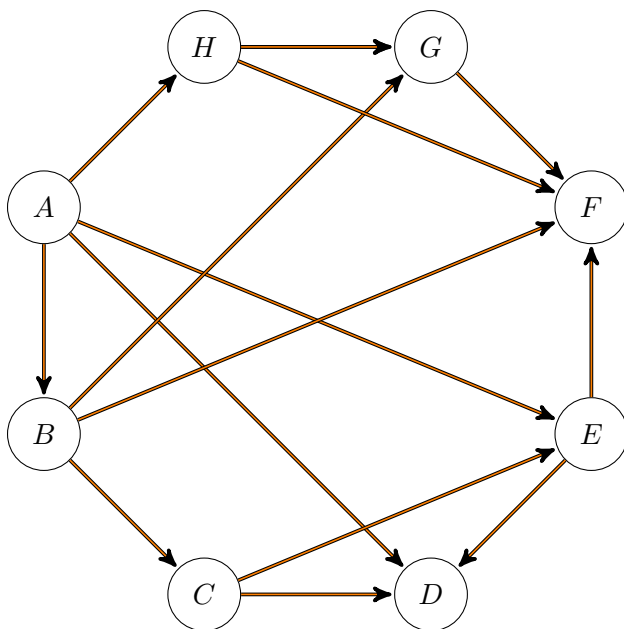
1. If you worked in a group, list the members of the group. Otherwise, write “I did not work in a group.”
2. If you received significant ideas about your solutions from anyone not in your group, list their names here. Otherwise, write “I did not consult anyone except my group members”.
3. List any resources besides the course material that you consulted in order to solve the material. If you did not consult anything, write “I did not consult any non-class materials.”

1. (10 pts.) Consider the following directed graph.



1. What are the sources and sinks of the graph?
2. Give one linearization of this graph.
3. How many linearization does this graph have?

2. (10 pts.) Consider the following directed graph.



1. Perform depth-first search with timing (DFS-with-timing) on the above graph; whenever there's a choice of vertices, pick the one that is alphabetically first: give the pre and post number of each vertex.

2. Draw the meta-graph of this graph, and give the vertices in each connected component.
3. (10 pts.) You are in charge of the United States Mint. The money-printing machine has developed a strange bug: it will only print a bill if you give it one first. If you give it a d -dollar bill, it is only willing to print bills of value $d^2 \bmod 400$ and $d^2 + 1 \bmod 400$. For example, if you give it a \$6 bill, it is willing to print \$36 and \$37 bills, and if you then give it a \$36-dollar bill, it is willing to print \$96 and \$97.
- You start out with only a \$1 bill to give the machine. Every time the machine prints a bill, you are allowed to give that bill back to the machine, and it will print new bills according to the rule described above. You want to know if there is a sequence of actions that will allow you to print a \$20 bill, starting from your \$1 bill. Model this task as a graph problem: give a precise definition of the graph (what are the vertices and edges) involved and state the specific question about this graph that needs to be answered. Give an algorithm to solve the stated problem and give the running time of your algorithm.