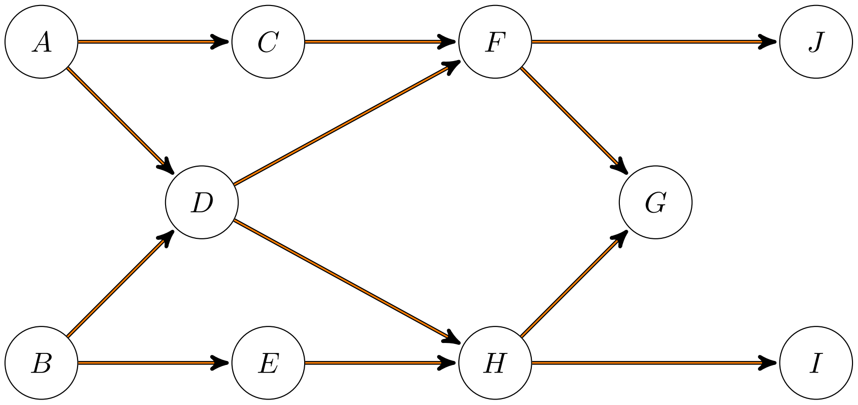
CMPSC 465 Data Structures & Algorithms
Spring 2022 Chunhao Wang and Mingfu Shao

Solution 05

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



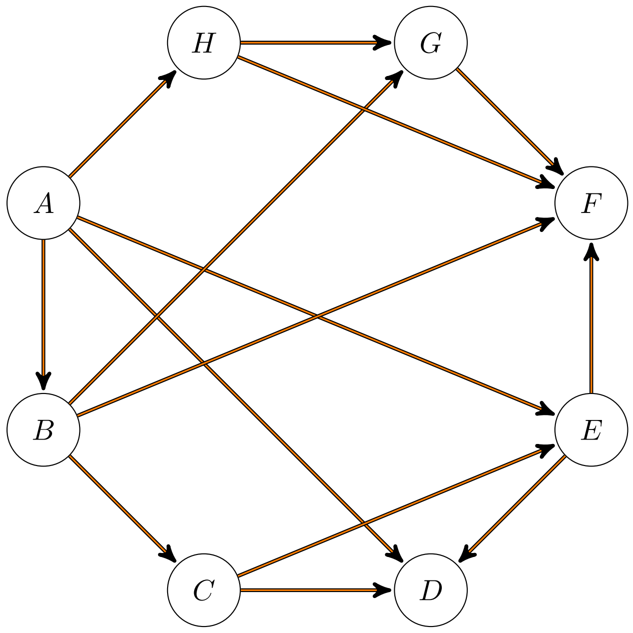
1. What are the sources and sinks of the graph?
2. Give one linearization of this graph.
3. How many linearization does this graph have?

Solution:

1. Sources of a graph do not have any in-edges. So, A and B are the sources. Sinks do not have any out-edges. So J, G and I are the sinks.
2. $(A, B, C, D, E, F, H, G, I, J)$ is one of linearizations.
3. It has 356 linearizations. Consider F and H; first assume that F is before H in the linearization. What are the possible situations between F and H in the linearization? There are 5 cases:
   (a) FH. Then A, B, C, D, E are before it and G, J, I are after it. It is not hard to calculate that the number of linearizations for the former is 16 and the number of linearization for the latter is 6.
   (b) FEH. Then A, B, C, D are before it and G, J, I are after it. The number of linearizations for the former is 5 and the number of linearization for the latter is 6.
   (c) FJH. Then A, B, C, D, E are before it and G, I are after it. The number of linearizations for the former is 16 and the number of linearization for the latter is 2.
   (d) $FEJH$. Then A, B, C, D are before it and G, I are after it. The number of linearizations for the former is 5 and the number of linearization for the latter is 2.
   (e) $FJEH$. Then A, B, C, D are before it and G, I are after it. The number of linearizations for the former is 5 and the number of linearization for the latter is 2.

   Summing them up gives $16 \cdot 6 + 5 \cdot 6 + 16 \cdot 2 + 5 \cdot 2 + 5 \cdot 2 = 178$. Don't forget this is under the assumption that F is before H, the other way around is symmetric. So the total number of linearization is $178 \cdot 2 = 356$.

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.

Solution:

1. A: 1,16
   B: 2,13
   C: 3,10
   D: 4,5
   E: 6,9
   F: 7,8
   G: 11,12
   H: 14,15
2. As there are no cycles in the graph, meta graph of the given graph is itself.

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 $d^2 \mod 400$ and $d^2+1 \mod 400$. 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.

Solution: $G = (V, E)$.

V: we have at most 400 possible bills (includes \$0 bill). We add 400 vertices $V = \{v_1, v_2, ..., v_{400}\}$ to represent these 400 possible bills (e.g. v_1 represents \$1 bill, v_5 represents \$5 bill, and v_{400} represent \$0 bill which can be generated by giving \$20).

E: we enumerate all vertices to add edges. For each vertices v_k, we have $i = k^2 \mod 400$ and $j = k^2 + 1 \mod 400$, so we add a directed edge from v_k to v_i and a directed edge from v_k to v_j.

Then the problem becomes if there is a path to reach v_{20} from a given vertex v_1 in the built graph G.

We can apply the *explore* algorithm start from v_1 to solve this problem. The running time for applying *explore* algorithm is $O(|V| + |E|)$. The running time for creating vertices and edges is $O(|V| + |E|)$. So the total running time is $O(|V| + |E|)$.

Rubrics:

Problem 1, 10pts

- 2 points : Provided the correct answer.
- 3 points : Provided a correct linearization.
- 5 points(Bonus) : Provided the correct answer.
- 1 point: I don't know how to answer this question

Problem 2, 10pts

- 6 points : Provided the correct pre an post numbers for all nodes.
- 3 points : Provided a correct pre and post number but didn't follow alphabetical order when there is a choice of vertices.
- 4 points : identified that metagraph is the given graph itself.(no need to draw the graph).
- 1 point: I don't know how to answer this question

Problem 3, 10pts

- 5 points: construct the graph correctly
  - 2 points : add vertices correctly
  - 3 points : add edges correctly
- 5 points: correct algorithm and running time analysis
  - 1 points : transfer the problem correctly (e.g. check if there is a path from v_1 to v_{20})
  - 3 points : apply explore algorithm
  - 1 points : appropriate running time analysis
- 1 point: I don't know how to answer this question