Assignment 2

CSCI 3110: Design and Analysis of Algorithms

Due May 28, 2019

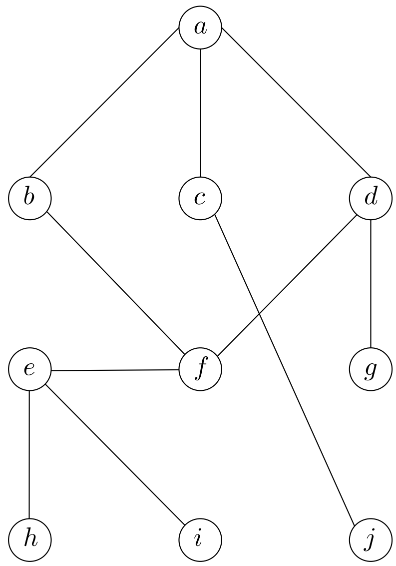
Banner ID: ____________________ Name: ______________________________

Banner ID: ____________________ Name: ______________________________

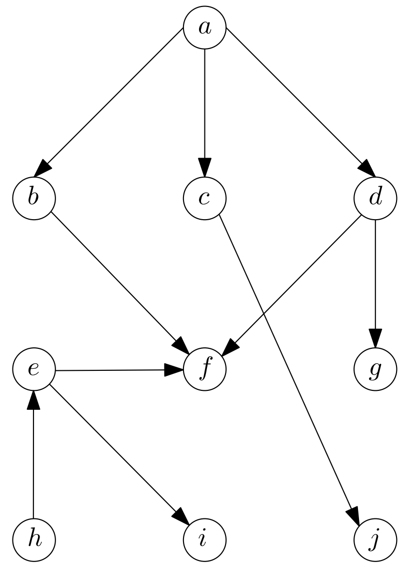
Banner ID: ____________________ Name: ______________________________

Assignments are due on the due date before class and have to include this cover page. Plagiarism in assignment answers will not be tolerated. By submitting their answers to this assignment, the authors named above declare that its content is their original work and that they did not use any sources for its preparation other than the class notes, the textbook, and ones explicitly acknowledged in the answers. Any suspected act of plagiarism will be reported to the Faculty's Academic Integrity Officer and possibly to the Senate Discipline Committee. The penalty for academic dishonesty may range from failing the course to expulsion from the university, in accordance with Dalhousie University's regulations regarding academic integrity.

1.



G_1



G_2

In this question we will try out some of the graph algorithms we have seen in class. Draw the annotated graphs that result from applying each of the following algorithms. You do not need to show the running states of the algorithms (i.e. the stacks, queues, etc) just the final result. Whenever you have a choice of visiting two vertices, visit them in alphabetic order (i.e. a before c, or u before v). Assume that edges are added to the stack or queue in the correct order so that they will be removed in alphabetic order. This also means that each of the algorithms should begin at a.

(a) (10 pts) Bipartitness Testing on graph G_1 to obtain the BFS tree. Is the graph bipartite? If not then report the odd cycle and remove the edge of any odd cycle with the alphabetically largest vertices until the graph is bipartite and run the algorithm again. Colour the vertices depending on whether they are at an even or odd level (black and white is fine) and draw the non-tree edges as dashed lines. What is the bipartitite partitioning?

(b) (10 pts) Topological Sorting on graph G_2 to obtain the DFS tree. Number the vertices and draw the non-tree edges as dashed lines. What is the topological ordering?

2. (10 pts) The square of a directed graph $G = (V, E)$ is the graph $G^2 = (V, E^2)$ such that $(u, v) \in E^2$ if and only if G contains a path with at most two edges between u and v. Give an efficient algorithm for computing the square of a graph in adjacency list format. Then, argue that your algorithm (i) terminates, (ii) is correct, and (iii) has the claimed running time.

3. Suppose that you are trapped with nothing but chalk in an enchanted labyrinth. The labyrinth consists of rooms connected by doors. One room contains the exit.

(a) (1 pt) How can you use the chalk to implement DFS?

(b) (2 pts) If you could not mark the doors of the labyrinth, could you still implement DFS by only marking rooms?

(c) (2 pts) If you could not mark the rooms of the labyrinth, could you implement DFS by only marking doors?

(d) (5 pts) Finally, suppose that doors between rooms can appear and disappear whenever you pass through a door. However, the doors do not change arbitrarily—there is always a path of unexplored rooms from your current position in the labyrinth to the exit (and you can see the mark of a room before passing through a door). Argue that DFS will still reach the exit.

(e) (BONUS: 5 pts) How long does DFS take in 4(d)?