CS 242: Data Structures and Algorithms

In-class part of Final Examination (December 17th, Thursday)

Name: ____________________ *Total Points: 150*

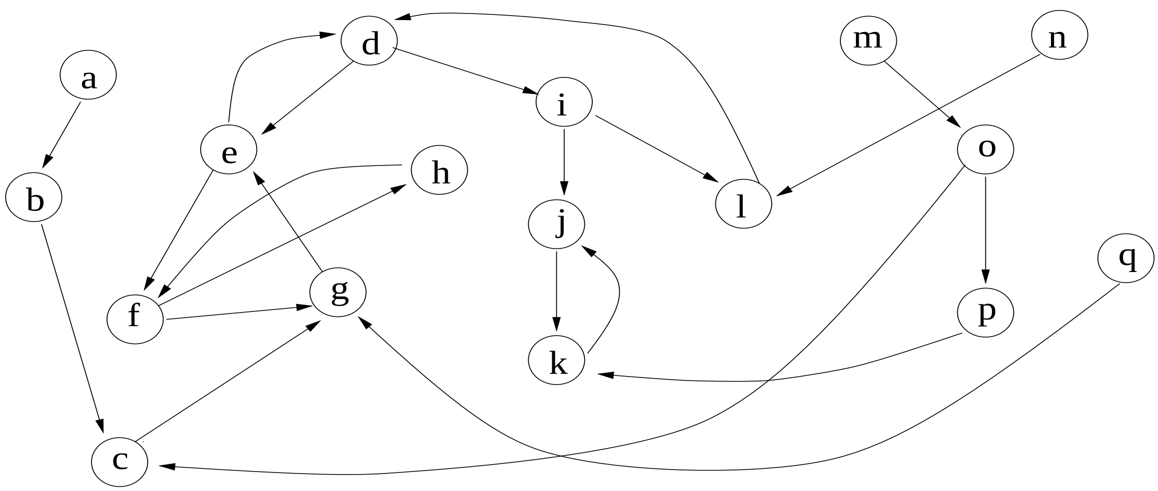
I. (30 Points) Circle **T** or **F** for each of the following statements to indicate whether the statements are true or false, respectively. If the statement is correct, briefly state why. If the statement is wrong, then correct it. The more content you provide in your justification or correction, the higher your grade, but be brief. One-sentence explanations should suffice.

T F By Case 2 of the Master Theorem, the solution to the recurrence $T(n) = 7T(n/2) + O(n^2)$ is $T(n) = \Theta(n^2 \lg n)$.

T F A 2-3-4-tree on n integer keys in the range 1 to n^2 can be constructed in $O(n)$ worst-case time.

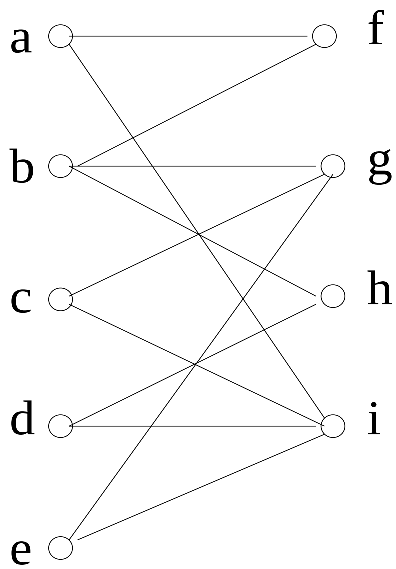
T F Kruskal's Algorithm for finding a minimum cost spanning tree is an example of a dynamic programming algorithm.

II. (30 Points) Consider the directed graph shown in the following figure. Perform a depth-first search on the graph starting at the vertex a. *For each vertex assume that the adjacency list is sorted alphabetically.* Your answer should consist of the resulting depth-first search forest. Each vertex in the forest should be labeled with the discovery and finishing times. You should show the cross edges, forward edges, and back edges in addition to the tree edges. Is it a singly-connected graph?



III. (30 Points) Consider again the directed graph shown in Question 2. Now perform breadth-first search starting from vertex a on this graph and show the resulting bfs tree with each vertex labeled by the shortest distance from the source vertex. Also show the cross edges and back edges. *For each vertex assume that the adjacency list is sorted alphabetically.*

IV. (30 Points) The graph shown below is an example of an **bipartite graph**. In a bipartite graph we can divide the set of vertices V into two subsets V_1 and V_2 such that all edges in the graph are from a vertex in V_1 to a vertex in V_2 or vice versa.



A **bipartite** graph

Provide an $O((m+n)$-time algorithm for recognizing an undirected bipartite graph. (*Hint. Start by running a breadth-first search on the graph. What properties does the BFS tree have if BFS is run on a bipartite graph?*)

V. (30 Points) Greedy Algorithms. Suppose you are given a set $\{x_1, x_2, \ldots, x_n\}$ of points on the real line.

1. (10 points) Describe an efficient algorithm that determines the smallest set of unit-length closed intervals that contains all of the given points.
2. (10 points) Analyze the running time of your algorithm.
3. (10 points) Argue that your algorithm is correct.
4. (**Extra Credit: 20 points**) Assume the points are all contained in the interval $[0, n]$. Can you solve the problem asymptotically faster? How?