I certify that the work submitted with this exam is mine and was generated in a manner consistent with this document, the course academic policy on the course website, and the Rutgers academic code. I also certify that I will not disclose this exam to others who are taking this course and who will take this course in the future without the authorization of the instructor.

Date: ____________________

Name (please print): ____________________

Signature: ____________________

Exam Time: 12 hours, 10 problems (12 pages, including this page)

- Write your name on this page and the last page, put your initials on the rest of the pages.
- If needed, use the last page to write your answer.
- Show your work to get partial credits.
- Show your rational if asked. Just giving an answer can't give you full credits.
- You may use any algorithms (procedures) that we learned in the class.
- Keep the answers as brief and clear as possible.

Name: ____________________________________

[Note: In this exam, both lg(x) and log(x) means $\log_2$(x)]
[Math facts: $\log a^b = b \log a$, $a^{\log b} = b^{\log a}$]

1. **(10 points) Asymptotic Growth of Functions**

List the 5 functions below in non-decreasing asymptotic order of growth:

$8^{(lg 2^n)}$ $\lg n^n$ $8^{\lg n}$ $\lg\lg(8^n)$ $8n + 8$

(1) __________ (2) __________ (3) __________ (4) __________ (5) __________
smallest largest

2. **(10 points) Properties of O, Ω, Θ**

Clues:

(1) $f_1(n) \in \Omega(8^{\lg n})$ (2) $f_2(n) \in \Omega(\lg\lg(8^n))$ (3) $f_3(n) \in O(\lg n^n)$
(4) $f_4(n) \in \Omega(8^{(lg 2^n)})$ (5) $f_5(n) \in \Theta(n+8)$

Circle TRUE (the statement must be always TRUE based on the clues above) or circle FALSE otherwise.

(a) $f_2(n) \in \Theta(f_5(n))$ TRUE FALSE

(b) $f_4(n) \in O(f_2(n))$ TRUE FALSE

(c) $f_4(n) \in O(f_5(n))$ TRUE FALSE

(d) $f_1(n) \in \Omega(f_3(n))$ TRUE FALSE

(e) $f_2(n) \in O(\lg n^n)$ TRUE FALSE

Name: ______________________________

3. **(10 points) Recursion Trees**
Use the recursion tree method to determine the asymptotic upper bound of T(n). T(n) satisfies the recurrence $T(n) = 2T(n-1) + c$, where c is a positive constant, and $T(0)=0$.

Name: ______________________________

4. **(10 points) Solving Recurrences**
(1) (5 points) Find a tight bound solution for the following recurrence:

$$T(n) = 2T(\frac{n}{2}) + c\,n^2 \;, \qquad T(1) = c \qquad (c \text{ is a positive constant})$$

That is, find a function g(n) such that $T(n) \in \Theta(g(n))$. For convenience, you may assume that n is a power of 2, i.e., $n=2^k$ for some positive integer k. Justify your answer. [Note: Read question 4-(2) first before writing your answer]

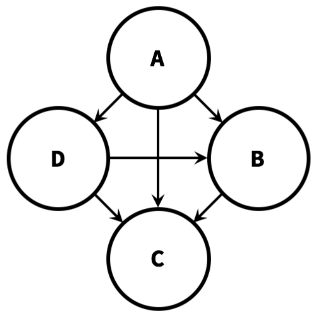
Name: ______________________________

(2) (5 points) Prove your answer in 4-(1) either using the iteration method or using the substitution method that we learned in our class.

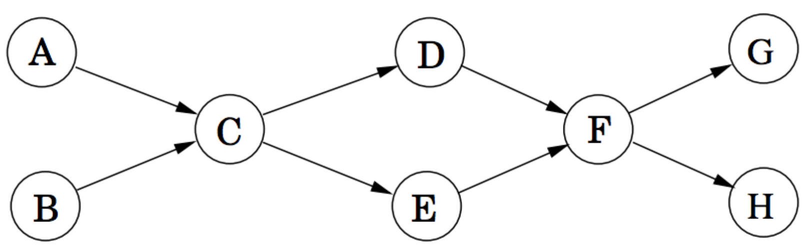
Name: ____

5. (10 points) **Graph Basics**

   (1) (4 points) For the following directed graph, provide the adjacency matrix representation and the adjacency list representation of the graph.



   (2) (3 points) DFS and its application: Run the DFS-based topological sort algorithm on the following graph. Whenever there is a choice of vertices, choose the one that is alphabetically first. Give the resulting topological ordering of the vertices.

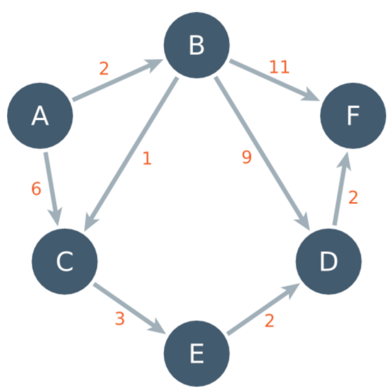


Name: ________________________________

(3) (3 points) BFS and its applications: You are given a directed graph G = (V, E), and a special vertex v. Give an algorithm based on BFS that determines in O(V +E) time whether v is part of a cycle.

6. (10 points) **Dijkstra's Algorithm**

Run Dijkstra's algorithm on the following graph, starting from vertex A. Whenever there are multiple choices of vertex at the same time, choose the one that is alphabetically first. You are expected to show how you initialize the graph, how you picked a vertex and update the d values at the each, and what is final shortest distance of each vertex from A.



Name: ________________________________

7. (10 points) **Minimum Spanning Tree**

We present an alternative algorithm to find the minimum spanning tree of an undirected graph:

1. T = null;
2. For each edge e, taken in arbitrary order:
3. T = T $\cup$ {e};
4. if T has a cycle c:
5. let e' be the maximum-weight edge on c;
6. T = T – {e'};
7. return T

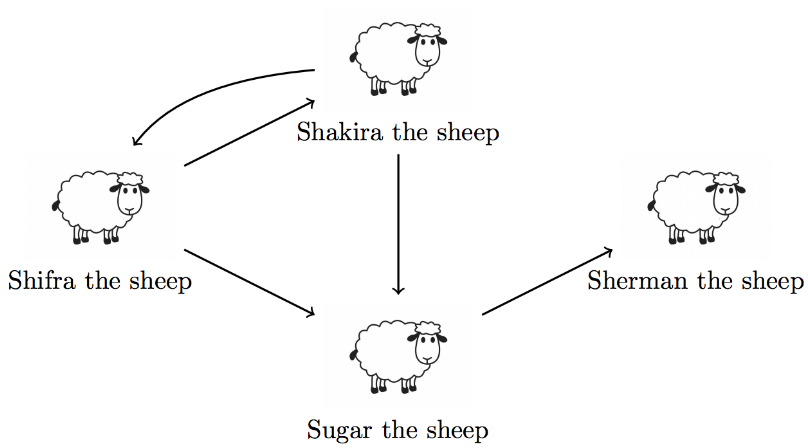
(1) (5 points) Explain that the final output T is a spanning tree of the graph. (A formal proof is not required).

(2) (5 points) Explain that the final output T is the *minimum* spanning tree of the graph. (A formal proof is not required).

Name: ______________________________

8. (10 points) **Strongly Connected Components**

You arrive on an island with n sheep. The sheep have developed a pretty sophisticated society, and have a social media platform called Baaaahtter (it's like Twitter but for sheep). Some sheep follow other sheep on this platform. Being sheep, they believe and repeat anything that they hear. That is, they will re-post anything that any sheep they are following said. We can represent this by a graph, where (a) → (b) means that (b) will re-post anything that (a) posted. For example, if the social dynamics on the island were:



the Sherman the Sheep follows Sugar the Sheep, and Sugar follows both Shakira and Shifra, and so on. This means that Sherman will re-post anything that Sugar posts, Sugar will re-post anything by Shifra and Shikira, and so on. (If there is a cycle then each sheep will only re-post a post once).

For the parts below, let G denote this graph on the n sheep. Let m denote the number of edges in G.

(1) (5 points) We call a sheep a source sheep if anything that they post eventually gets re-posted by every other sheep on the island. In the example above, both Shifra and Shakira are source sheep. Show that (a) all source sheep are in the same strongly connected component of G, and (b) every sheep in that component is a source sheep. (We are expecting a short but rigorous proof for both claims)

Name: ____

(2) (5 points) Suppose that there is at least one source sheep. Give an algorithm that runs in time $O(n + m)$ and finds a source sheep. You may use any algorithm we have seen in class as a subroutine. (We are expecting pseudocode or a description of your algorithm; a short justification that it is correct and a short justification of the runtime)

Name: ______________________________

9. (10 points) **Max-flow Min-cut**
Let G = (V, E) be a directed graph, it has a source vertex s, and a target vertex t. Each edge of the graph has a positive capacity c. Suppose someone hands you an arbitrary flow, i.e., each edge is assigned a flow number f. Describe fast and simple algorithms to answer the following questions:

(1) (3 points) Is the flow a feasible (s, t)-flow in G?

(2) (3 points) Is the flow a maximum (s, t)-flow in G?

(3) (4 points) Is the flow the unique maximum (s,t)-flow in G?

Name: ________________________________

10. (10 points) **Greedy Algorithms**
As a teaching administrator of the department, your responsibility is to schedule the classes for a particular classroom. Suppose there are *n* classes, each class *i* is represented by its start time and finishing time $[s_i, f_i]$, and we say that two classes i and j are non-conflicting if they do not overlap in time (i.e., $s_i \geq f_j$ or $s_j \geq f_i$). You want to schedule as many classes for the classroom as possible, but the scheduled classes should be non-conflicting. Develop an algorithm so that you can select the maximum number of classes for the classroom. (We are expecting either pseudocode or language description of your algorithm)