Name ____

CSCI 332, Fall 2025

Exam 1 Practice 1

Note that this exam has four sections. They are:

1. Stable Matching (30 points)
2. Algorithm Analysis (30 points)
3. Graph Algorithms (30 points)
4. Greedy Graph Algorithms (10 points)

You may use a double sided 3x5 handwritten notecard of notes during the test but no other resources. If you need more space than what is given, develop your solution on scratch paper before copying your final answer to the exam paper.

Good luck!

Section 1 (Stable Matching)

1. (5 points) The input to the *stable matching problem* is a set of n men, a set of n women, and preference lists ranking each man for each woman and each woman for each man. Describe the desired output as precisely as you can.

2. A matching for the following preference lists for 4 men and 4 women is given by the capitalized men and women.

$$
\begin{array}{ll}
m_1\colon\ w_3, W_1, w_2, w_4 & w_1\colon\ m_4, m_3, m_2, M_1 \\
m_2\colon\ W_4, w_1, w_2, w_3 & w_2\colon\ M_4, m_3, m_2, m_1 \\
m_3\colon\ w_2, w_1, W_3, w_4 & w_3\colon\ m_1, m_2, m_4, M_3 \\
m_4\colon\ w_1, W_2, w_3, w_4 & w_4\colon\ m_1, m_3, m_4, M_2
\end{array}
$$

3. (3 points) Is this matching stable? Yes or no

4. (2 points) How do you know?

5. (5 points) This is the same preference list for 4 men and 4 women given above.

$$
\begin{array}{ll}
m_1\colon\ w_3, w_1, w_2, w_4 & w_1\colon\ m_4, m_3, m_2, m_1 \\
m_2\colon\ w_4, w_1, w_2, w_3 & w_2\colon\ m_4, m_3, m_2, m_1 \\
m_3\colon\ w_2, w_1, w_3, w_4 & w_3\colon\ m_1, m_2, m_4, m_3 \\
m_4\colon\ w_3, w_2, w_1, w_4 & w_4\colon\ m_1, m_3, m_4, m_2
\end{array}
$$

   What is the outcome of the Gale-Shapley algorithm on this input? Assume that men propose to women.

6. (5 points) True or false? In every instance of the Stable Matching Problem, there is a stable matching containing a pair (m, w) such that m is ranked first on the preference list of w and w is ranked first on the preference list of m.

Section 2 (Algorithm Analysis)

7. (10 points) For each of the following statements, circle T if it is true and F if it is false.

   - 2^n is $O(n!)$: T or F
   - n^3 is $\Omega(n^2)$: T or F
   - $n \log n$ is $\Theta(n^2)$: T or F
   - There is an algorithm with worst-case runtime that is $\Omega(n^2)$ and best-case runtime that is $\Omega(n \log n)$: T or F
   - There is an algorithm with worst-case runtime that is $O(n^2)$ and best-case runtime that is $\Theta(n^3)$: T or F

8. (5 points) Give a function $f(n)$ such that the worst-case runtime of the following algorithm is $\Theta(f(n))$.

```
Algorithm-1(array A of length n):
  Result = 0
  For every third i in 1 to n:
      For j in 1 to i:
          Set Result to A[j] times Result
  Return Result
```

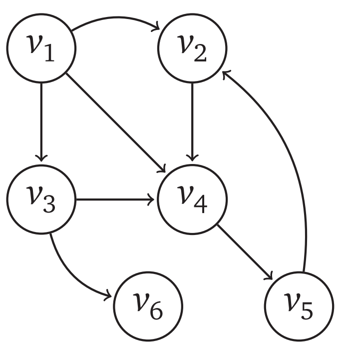
9. (5 points) For the same algorithm above, give a function $g(n)$ such that the best-case runtime of the algorithm is $\Theta(g(n))$.

Section 3 (Graph Algorithms)

10. (5 points) Finish the following algorithm to perform breadth-first search on an input graph G.

```
BFS(graph G and starting node v):
    Set all nodes' layer to null, except set v's to 0
    Set L = 0
    While there is a node whose layer is null:
```

11. (5 points) Give every topological ordering of the following DAG.



12. (5 points) Fill in the blanks to complete the proof that any DAG has a topological ordering.

Proof: Universal declaration: Let G be ______________________________________.

Inductive hypothesis (for this proof, use "fewer nodes than" for your definition of "smaller than"):

There are two cases to consider:

- G has 1 node. Then,

- G has $n > 1$ nodes.

Since G was an arbitrary DAG, and the claim holds in all cases, every DAG has a topological order. □

Section 4 (Greedy Graph Algorithms)

Recall Dijkstra's algorithm to compute the shortest path distance from one node s to all other nodes in a directed graph with non-negative edge weights:

```
Dijkstra(directed graph G = (V, E) with non-negative edge weights; node s)
  Let S be the set of explored nodes
  For each u ∈ S, we store a distance d(u)
  Initially S = {s} and d(s) = 0
  While S ≠ V:
      Select a node v ∉ S with at least one edge from S for which
      d'(v) = min_{e=(u,v):u∈S} d(u) + ℓ_e is as small as possible
      Add v to S and define d(v) = d'(v)
```

13. (3 points) How many times *exactly* does the while loop run? (This should be a function of m, the number of edges, and/or n, the number of nodes.)

14. (3 points) Suppose that we implement the while loop naively by examining every $v \notin S$, computing $\min_{e=(u,v):u\in S} d(u) + \ell_e$, and then using those values to find the next node to add to S. Describe a a worst-case input for this implementation.

15. (4 points) Assuming this naive implementation of the while loop, give a $f(n)$ such that the worst-case runtime of Dijkstra's algorithm on a graph with n nodes is $\Theta(f(n))$.