Name ____

CSCI 332, Fall 2024
Final Exam—Practice 1

Note that this exam has six sections. They are:

1. Stable Matching
2. Algorithm Analysis
3. Graph Algorithms (note that this document currently ends here)
4. Divide and Conquer Algorithms
5. Greedy Algorithms
6. Dynamic Programming Algorithms

You may use a 3x5 handwritten notecard of notes during the test both 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)

Recall the *stable matching problem*: given n men, n women, and preference lists ranking each man for each woman and each woman for each man, find a matching that contains no unstable pairs.

1. (2 points) How many matchings (not necessarily stable) are there, as a function of n?

2. (5 points) Decide whether the following statement is true or false. If it is true, give a short explanation. If it is false, give a counterexample.

   True or false? There is an instance of the Stable Matching Problem with a stable matching containing a pair (m, w) *such that* m *is ranked last on the preference list of* w *and* w *is ranked first on the preference list of* m.

3. (5 points) Consider the following preference lists for 4 men and 4 women.

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

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

4. (3 points) Given the above preference, give another stable matching that is not the output of Gale-Shapley. Here is another copy of the preference lists in case it is helpful.

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

5. (5 points) Describe a worst-case input, in terms of runtime, for the Gale-Shapley algorithm.

6. (3 points) Give a function $f(n)$ such that the worst-case runtime of the Gale-Shapley algorithm on an input of size n is $\Theta(n)$.

7. (2 points) Describe a best-case input, in terms of runtime, for the Gale-Shapley algorithm.

Section 2 (Algorithm Analysis)

8. (6 points) For each of the following statements, circle T if it is true and F if it is false.

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

9. (4 points) Suppose you have algorithms with the following runtimes. (Assume these are exact running times as a function of the input size n, not a asymptotic running times.) How much slower do these algorithms get when you double the input size? (You can find this by dividing the runtime on $2n$ by the runtime on n). You should simplify your answer as much as possible. If n values do not cancel, you can describe what the value approaches as n gets large.

   (a) $3n^2$

   (b) $\log n$

10. (3 points) In words, what is the definition of the worst-case runtime for an algorithm?

11. (3 points) Give a function $f(n)$ such that the worst-case runtime of the following algorithm is $\Theta(f(n))$. Recall that $\lfloor x \rfloor$ takes the *floor* of x, meaning that it rounds down to the nearest integer.

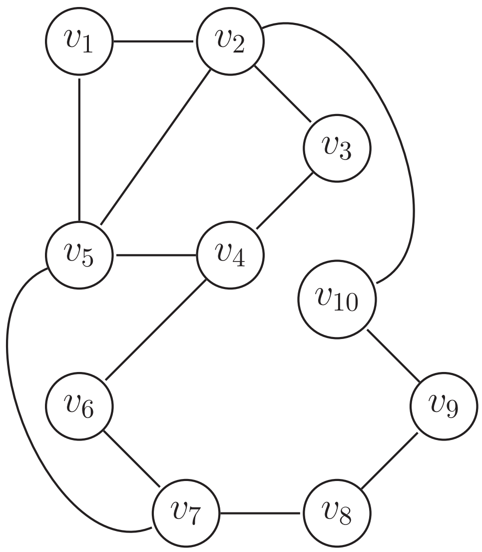
```
Algorithm-1(array A of length n):
  Result = 0
  For i in 1 to n:
      For j in i + ⌊n/2⌋ to n:
          Add A[j] to Result
  Return Result
```

12. (4 points) Give a function $f(n)$ such that the worst-case runtime of the following algorithm is $\Theta(f(n))$.

```
Algorithm-2(array A of length n):
  Result = 0
  For i in 1 to n:
      For j in 2^i to n:
          Add A[j] to Result
  Return Result
```

Section 3 (Graph Algorithms)

13. (3 points) Given the following undirected graph, give the nodes contained in the sets for each layer that would be generated by running breadth-first search starting at v_1. You may not need all of layers L_0 through L_6.



$L_0 =$

$L_1 =$

$L_2 =$

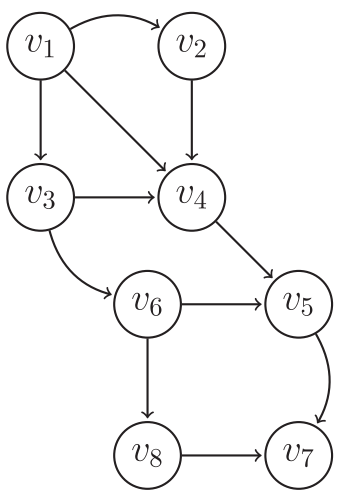
$L_3 =$

$L_4 =$

$L_5 =$

$L_6 =$

14. (3 points) Given the following directed graph, give a topological ordering of the nodes.



15. (3 points) Decide whether the following statement is true or false. If it is true, just write True. If it is false, give a counterexample.

    True or false? Every connected tree has exactly $n - 1$ edges.

16. (3 points) Finish the following algorithm to find a topological ordering of a graph. Assume that the input is a directed graph with no cycles.

```
Topo-Order(length n list of nodes and length m list of edges):
  T = [], an empty list to hold the topological order of the nodes
  While there is a node that hasn't been added to T yet:
        Look through edge list to find v, a node with no incoming edges
```

17. (8 points) Fill in the blanks to complete the proof that in any binary tree the number of nodes with two children is exactly one less than the number of leaves. (Recall that a binary tree is a rooted tree where every node has no more than two children.)

Proof: Let T be a ____.

Inductive hypothesis: Suppose that every binary tree with ____________________

____

____

There are two cases to consider:

- T has 1 node. Then, ____

  ____

  ____.

- T has $n > 1$ nodes. In this case, if we remove a leaf node v and form T', we have formed a tree with fewer nodes than T in which each node has at most two children, and by the inductive hypothesis, the number of nodes with two children is exactly one less than the number of leaves. Now we must consider two cases:

  – Suppose v was the only child of its parent u in T. ____________________

    ____

    ____

    ____

    ____

    ____.

  – Suppose v was one of two children of its parent u in T. ______________

    ____

    ____

    ____

    ____

    ____.

$\square$