University of New Mexico
Department of Computer Science

Final Examination

CS 362 Data Structures and Algorithms
Spring, 2025

Name:
Email:

Directions:

- This exam lasts 120 minutes. It is closed book and notes, and no electronic devices are permitted. However, you are allowed to use 2 pages of handwritten "cheat sheets"
- *Show your work!* You will not get full credit, if we cannot figure out how you arrived at your answer.
- Write your solution in the space provided for the corresponding problem.

Question	Points	Score	Grader
1	20		
2	20		
3	20		
4	20		
5	20		
Total	100		

1. **Short Answer (4 points each)**

   Unless specified otherwise, answer using *simplest* Θ *notation.*

   (a) Solution to the recurrence: $T(n) = 2T(n/2) + n^2$.

   (b) Solution to the recurrence: $f(n) = 6f(n-1) - 9f(n-2) + 3^n$, in big-O notation.

   (c) Worst case time to compute a minimum spanning tree on a graph with n nodes and $\Theta(n^2)$ edges using Prim's algorithm.

   (d) Expected number of pairs of balls that fall in the same bin when n balls are dropped independently and uniformly at random in n bins.

   (e) $3n$ fish are split into n bowls, each containing 3 fish. Each fish becomes a cannibal independently with probability $1/3$, and eats all non-cannibal fish in their bowl. How many non-cannibal fish remain alive in expectation? Answer exactly, not in asymptotic notation.

2. **Induction** (20 points) Consider a graph $G = (V, E)$. Call an edge *satisfied* if both endpoints of the edge are different colors. Prove that there is a way to color the nodes of G with 2 colors, such that at least half the edges are satisfied.

   Prove this by induction on the number of nodes in G. Hint: In the IS think about how to make G smaller so that you can use the IH.

3. **Amortized Analysis**

In class we discussed a binary counter with an INCREMENT operation taking $O(1)$ amortized time. Now we want to include a RESET operation that sets all the bits to 0. Below are the algorithms for INCREMENT and RESET. They maintain a bit array B, and a number m that is the largest index in B set to 1.

Algorithm 1 INCREMENT

```
i ← 0
while B[i] = 1 do
    B[i] ← 0
    i ← i + 1
end while
B[i] ← 1
if i > m then
    m ← i
end if
```

Algorithm 2 RESET

```
for i ← 0 to m do
    B[i] ← 0
end for
```

Give the following costs as a function of n, the total number of operations on the counter.

(a) (2 points) Worst-case time of a call to INCREMENT?

(b) (2 points) Worst-case run time of a call to RESET?

(16 points) Now prove that in an arbitrary sequence of calls to INCREMENT and RESET, each call has amortized cost O(1). Hint: Use the accounting method. Save up dollars during INCREMENT for future calls to RESET.

3. **Amortized Analysis, continued.**

4. **MAX-TRIANGLES** For a given set of nodes S in a graph G, call a triangle of G *inside* S if all 3 edges of the triangle have both endpoints in S.

   In the MAX-TRIANGLES problem, you are given a graph $G = (V, E)$, and a number x. You must output the *maximum* number of triangles inside any set $S \subseteq V$ that is of size x.

   (a) (10 points) Prove that MAX-TRIANGLES is NP-Hard by a reduction from one of the following: 3-SAT, VERTEX-COVER, CLIQUE, SUBGRAPH-ISOMORPHISM, INDEPENDENT-SET, 3-COLORABLE, HAMILTONIAN-CYCLE, or TSP.

(b) (10 points) Consider the randomized algorithm that picks a subset S of size x, uniformly at random from all subsets of size x. Compute the expected number of triangles inside S. Let T be the total number of triangles in the graph, and $n = |V|$. Give your answer as a function of t, x and n and m. Hint: Use indicator random variables and linearity of expectation. You can give your final answer using choose (i.e. $\binom{x}{y}$) notation.

5. **Cake Cutting**

(20 points) You've purchased a rectangular cake for graduation that's m inches long and n inches wide. You plan to cut it into smaller rectangular pieces with integer dimensions to re-sell. You have a price list $P[x, y]$ that shows how much you can sell any x by y inch piece for. These prices vary based on factors like appearance and demand, with no consistent pattern. You can only make horizontal or vertical cuts that go completely through any rectangular piece.

Create a dynamic programming solution that finds the maximum possible total selling price when you cut up the original m by n cake in the optimal way. Be sure to include the following:

(a) English description of the subproblems (e.g. "minimum edit distance of first i characters of string A and first j characters of string B").

(b) Math description of the recurrence (e.g. "Base case: e(0,i) = e(i,0) = i; Else: e(i,j) = min(e(i,j-1), e(i-1,j), e(i-1,j-1) + 1 - I(A[i]=B[j])").

(c) How to compute the final answer (e.g. "Return e(m,n)").

(d) How to solve the subproblems and analysis of your runtime (e.g. "Fill in a table left to right, top down. Runtime is $O(nm)$")