University of New Mexico
Department of Computer Science

Midterm Examination

CS 561 Data Structures and Algorithms
Fall, 2024

Name:
Email:

Directions:

- This exam lasts 75 minutes. It is closed book and notes, and no electronic devices are permitted. However, you are allowed to use 2 pages of "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)**

   Answer the following using *simplest possible* Θ *notation.*

   (a) Expected number of items at the $(1/2)\log n$ level of a skip list containing n items?

   (b) Assume n items have been inserted into a count-min sketch with k hash functions and m counters. What is the time to get a estimate for the number of times some item has been seen?

   (c) Solution to the recurrence: $T(n) = 4T(n/4) + n$

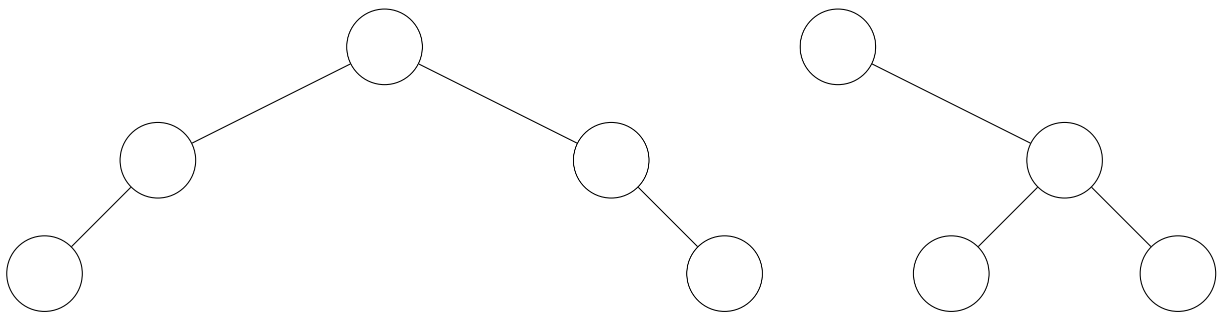
   (d) Solution to the recurrence: $T(n) = 8T(n/2) + \sqrt{n}$

   (e) Solution to the recurrence: $f(n) = 4f(n-1) - 3f(n-2) + 3^n$ (answer in big-O)

2. **Induction (20 points)**

In a *leaf-balanced binary tree*, any node with 2 children has the same number of leaves in the sub-trees rooted at both children.

A full binary tree is leaf-balanced; below are two other examples.



(20 points) Let n be the number of nodes in the tree. Prove by induction on n that the number of leaf nodes in a leaf-balanced binary tree is always a power of 2.

3. **Probability**

On Halloween, each house in your neighborhood gives out 3 types of candy: (M)ars, (R)eese's, and (C)andy Corn. At each house, you get 10 pieces of candy, each piece sampled independently from the 3 types with uniform probability. You assign a value to pieces as follows: 30 (M), 3 (R), and 0 (C).

(a) (5 points) What is the expected value of the candy you get from one house?

(b) (5 points) At one house, you get a value that is 10 times the expected value you computed above. Use Markov's inequality to bound the probability that your friend will get that value or higher at the next house.

(c) (5 points) At a spooky house at the end of your block, they only have 30 pieces of candy left, 10 of each type. When they give out candy, they select 10 pieces random from their supply. What is the expected value of the candy you get from this house? How does it compare to your answer in part (a)?

(d) (5 points) At a decrepit house at the other end of your block, they have n pieces of candy in a large bag, exactly one of which is a candy corn. The old man at this house selects 3 pieces of candy from the bag for you. Use a union bound to upper bound the probability that you receive a candy corn. Leave your answer as a sum of fractions.

4. **The Creaky Staircase**

On a dark and quiet fall night, you find yourself climbing a creaky staircase with n stairs. Every stair, $i \in [1, n]$ has a creakiness value (or cost) c_i. In each step, you can go up either 2 or 3 stairs, and your goal is to get to the top of the staircase in a way that minimizes the sum of costs for all the stairs you visit. The bottom and top of the staircase are not creaky and so have no cost. On the last stair, either step size will take you to the top.

For example, if $n = 4$ and the costs are $[2, 1, 5, 8]$, you can first go up 2 stairs for a cost of 1, and then go up 3 stairs for a cost of 0, for a total cost of 1.

(a) (4 points) Consider a greedy algorithm for this problem which always decides the next move based on which of the two choices has least cost. Give an example where this greedy algorithm is not optimal.

(b) (12 points) Now write a recurrence relation to solve this problem. Don't forget to first define the function in words whose solutions enable solving the big problem.

(c) (4 points) Describe a dynamic program to solve the problem using your recurrence. What are the dimensions of your table? How do you fill it in? What is the final value returned? What is the runtime of your algorithm?

5. **Game of NIM**

The ancient game of *NIM* is played by two players who alternate taking any positive number of stones[1] from one of 3 piles. The person taking the last stone loses. An example game starting with piles of size 13, 9 and 1 is below. Here, the first move of Player 1 is to take the single stone in pile 1, and player 1 eventually wins.

Player Turn	1	2	1	2	1	2
Stones left	(13, 9, 1)	(13, 9, 0)	(13, 2, 0)	(2, 2, 0)	(1, 2, 0)	(1, 0, 0)

In this problem, you will write a dynamic program to determine if Player 1 can force a win for a given input specifying the sizes of the 3 piles.

(a) (2 points) Describe in words a function whose solutions for smaller problems will help you solve the big problem.

(b) (13 points) Write a recurrence relation for the dynamic program using the function you described above.

[1] knuckle bones, dried spiders, or (perhaps apocryphally) candy corns are rumored replacements for stones.

(c) (5 points) Describe a dynamic program to solve the problem for any initial input consisting of x_1, x_2, x_3 stones in the piles. What are the dimensions of your table? How do you fill it in? What is the final value returned? What is the runtime of your algorithm if $x_1 = x_2 = x_3 = n$?