University of New Mexico
Department of Computer Science

Midterm Examination

CS 561 Data Structures and Algorithms
Fall, 2019

Name:
Email:

- 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 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.
- If any question is unclear, ask for clarification.

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

1. **Short Answer**

   Answer the following questions using the simplest possible *theta* notation. Assume in recurrences that $f(n)$ is $\theta(1)$ for constant values of n.

   (a) $\sum_{i=1}^{n} \log i$

   (b) $\sum_{i=1}^{n} (1/3)^i$

   (c) Solution to the following recurrence $T(n) = 8T(n/2) + n^2$

   (d) Solution to the following recurrence relation: $f(n) = 4f(n-1) - 3f(n-2) + 3^n$.

   (e) Consider a certain operation *OP*. Say that over n calls to OP, the last call always takes $\Theta(n)$ time and that all previous calls take $\Theta(1)$ time. What is the amortized cost of OP?

2. **Probability and Expectation**

In the homework, you colored nodes in a network. Now you will color *edges*. You are given a network that is 4-regular in that each node has exactly 4 edges touching it. The graph has n nodes and $2n$ edges. Each edge is colored independently and uniformly at random with one of 2 colors: color 1 and color 2. A node is *unsatisfied* if all edges touching it are colored with color 1.

(a) (3 points) What is the probability that a fixed node is unsatisfied? Hint: Remember that the 4 edges touching the node are colored independently.

(b) (4 points) What is the expected number of unsatisfied nodes?

(c) (4 points) Use Markov's inequality to get an upper bound on the probability that at least $n/2$ nodes are unsatisfied.

(d) (4 points) Fix any set of 10 nodes. Use a union bound to get an upper bound on the probability that any node in this subset is unsatisfied.

(e) (5 points) Call an edge *unsatisfied* if both nodes that edge touches are unsatisfied. What is the expected number of unsatisfied edges? (Recall that there are $2n$ total edges.)

3. **Dynamic Programming**

(a) (8 points) In the bagel problem, there are 3 types of boxes that fit quantities of x_1, x_2 and x_3 bagels respectively. Additionally, the boxes have costs of 1, 1 and 2 respectively. Let $c(n)$ be the minimum cost way to box n bagels if this is possible or ∞ otherwise. Write a recurrence relation for $c(n)$.

(b) (2 points) If you design a dynamic program based on this recurrence to compute $c(n)$, what is the runtime of your algorithm?

(c) (8 points) In the muffin problem, there are 3 types of boxes that fit quantities of x_1, x_2 and x_3 muffins respectively. Your boss insists that any time a box is used, it must be completely filled with muffins. The goal is to finding a boxing of muffins that minimizes the number of unboxed muffins. Let $f(n)$ be the minimum number of unboxed muffins, when boxing n muffins. Write a recurrence relation for $f(n)$.

(d) (2 points) Will a greedy algorithm that always puts remaining muffins in the largest box possible solve this problem? Give a proof or counter-example.

4. **Probability and Recurrences**

There are two bins. Bin 1 initially has 3 white balls and 1 red ball. Bin 2 has 4 white balls. In every round, a ball is selected uniformly at random from each bin and these two balls are swapped.

Let p_k be the probability that the red ball is in bin 1 at the beginning of the k-th round.

(a) (8 points) Write a recurrence relation for p_k.

(b) (8 points) Use the guess and check and proof by induction to solve this recurrence. Don't forget to label BC, IH and IS and clearly say where you are using the IH. Hint: Compute the first few values of p_k to spot the pattern.

(c) (4 points) What is the expected number of rounds that the red ball is in bin 1 over m rounds?

5. **Dynamic Programming II** (20 points) You are given a chocolate bar of length n and width 1 that is made up of n chunks of width and length 1. Each chunk $1 \leq i \leq n$ has some positive value v_i.

   You must break the bar into exactly k pieces to share with your friends, where each piece consists of some number of contiguous, unbroken chunks. Your (greedy) friends then each get to chose their pieces first, and you get the last piece. Thus, your goal is to do the breaking in such a way that you maximize the value of the minimum value piece. The value of a piece is simply the sum of the value of the chunks in that piece.

   Give a dynamic program to solve this problem. What is the runtime of your algorithm? Note: This is a recent Amazon interview question.