University of New Mexico
Department of Computer Science

Midterm Examination

CS 561 Data Structures and Algorithms
Fall, 2018

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 as usual, that $f(n)$ is $\theta(1)$ for constant values of n.

   (a) Expected runtime of randomized QuickSort on a list with n elements?

   (b) Expected number of items at the 2-nd level of a skip list containing n items?

   (c) Expected time to do a search in a skip list containing n items?

   (d) Solution to the following recurrence $T(n) = 4T(n/2) + \sqrt{n}$

   (e) Solution to the following recurrence relation: $f(n) = 4f(n-1) - 2f(n-2)$.

2. **Probability and Expectation**

(a) (3 points) Each of your n friends starts out with a token. Each token is one of $n/2$ possible colors, and exactly 2 friends have each color (n is even). You put all the tokens in a hat, shake it, and then return the tokens to your friends in a random order. What is the expected number of friends who get back their original color?

(b) (5 points) Use Markov's inequality to get an upper bound on the probability that at least $n/10$ of your friends get their original color back

(c) (6 points) Fix a subset of $n/10$ of your friends. Use a union bound to get an upper bound on the probability that someone in this subset gets their original color back.

(d) (6 points) There are $n/2$ pairs of friends who start out with the same color. Among these $n/2$ pairs, what is the expected number of such pairs where both people in the pair get back their original color?

3. **Dynamic Programming**

(a) (6 points) A lazy thief repeatedly robs the same bank. To avoid capture, he decides to never rob the bank fewer than 10 days prior to the last robbery. He has obtained information, for the next n days, on the amount of money b_i that is held at the bank on day i. Let $r(i)$ be the maximum amount of revenue the thief can safely obtain. Give a recurrence relation for $r(i)$.

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

(c) (10 points) Now on each day, there is an integer value giving the amount of work w_i that the thief must perform to rob the bank on that day (due to the amount of security on that day). Also there is an additional constraint that the sum of work the thief ever performs is less than some value W. Let $r(i, j)$ be the maximum amount of revenue obtainable on days 1 through i, with at most j total work. Give a recurrence relation for $r(i, j)$.

(d) (2 points) If you design a dynamic program based on this recurrence, what is the runtime of your algorithm?

4. **Amortized Analysis**

(a) (10 points) Consider a (distributed) data structure that uses computational puzzles to control membership. There is only one operation provided: JOIN. This allows nodes to join the system by solving a puzzle with a computational cost of 1. In addition, there are periodic purges where all nodes in the system must again solve a puzzle with a computational cost of 1. A purge occurs when the number of JOINs since the last purge is 1/3 the size of the system after the last purge. What is the amortized cost of the JOIN operation? Show your work. Hint: Use the accounting method, and make sure you tax JOIN enough to pay for all puzzle costs (at entrance and during all purges).

(b) (10 points) Now what if nodes can leave, and purges happen when the number of join and leaves since the last purge is $1/3$ the size of the system after the last purge? Again there is only one operation that is provided by the system: JOIN (nodes leave at will). What is now the amortized cost of JOIN? Hint: Again use the accounting method. You will need to raise the taxes on JOIN.

5. **Uniform Sampling**

You are watching a stream of packets go by one at a time and want to sample k distinct packets. You face several problems

- You only have room to hold k packets at a time
- You do not know the total number of packets in advance
- If you don't save a packet as it goes by, it is gone forever

(a) (10 points) Design and prove correct an algorithm that ensures that whenever the packet stream stops, you have a set of k packets chosen uniformly at random from the entire stream. If the stream has less than k packets, you should hold all of these packets.

(b) (10 points) At time n, what is the expected number of packets that are ever held by your algorithm, even if only temporarily?