University of New Mexico
Department of Computer Science

Midterm Examination

CS 561 Data Structures and Algorithms
Fall, 2016

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 a 1 page "cheat sheet"
- *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 Bucket Sort on a list with n elements that are each independent random variables between 0 and 1?

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

   (c) Amount of time required to store n items in a count min sketch when ϵ and the probability of error (i.e. it is note true that $m(i, T) \leq count(i, T) + \epsilon T$) are both constants?

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

   (e) Solution to the following recurrence relation: $f(n) = 5f(n-1) - 6f(n-2)$.

2. **Probability and Expectation**

Imagine that you are buying pizza for n friends. You ask each friend to give you a dollar coin. When you get to the pizza place, it is closed, so you return the coins to your friends in a random order.

(a) (3 points) Consider a single friend. What is the probability that they get back their own coin?

(b) (3 points) What is the expected number of friends who will get back their own coins?

(c) (4 points) Use Markov's inequality to get an upper bound on the probability that at least 2 of your friends will get their coin back

(d) (5 points) What is the expected number of pairs of friends i and j, such that friend i gets j's coin and friend j gets i's coin?

(e) (5 points) Say that you lose a random subset of $n/2$ coins on the way back, and you randomly distribute the remaining coins to a random subset of $n/2$ of your friends. Now use a union bound to get an upper bound on the probability that at least one of your friends gets back their own coin back.

3. **Dynamic Programming**

(a) (6 points) You are designing an algorithm to build billboards at n possible locations on a highway. For all i between 1 and n, you know that location i will give you revenue r_i. Unfortunately, laws require that all billboards need to be at least 5 locations apart from each other. Let $m(i)$ be the maximum amount of revenue obtainable using locations 1 through i. Give a recurrence relation for $m(i)$.

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

(c) (10 points) Now you have an additional constraint that the total number of billboards you can build is at most x, for some given parameter x. Let $m(i, j)$ be the maximum amount of revenue obtainable using locations 1 through i, and j billboards. Give a recurrence relation for $m(i, j)$.

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

4. **Amortized Analysis**

(a) (6 points) Assume you are using the Union-Find data structure discussed in class to determine if there is a path from a node s to a node t in a graph with m edges and n nodes. Recall that the amortized costs of Union, Find-Set and Make-Set are all $O(\log^* n)$. Briefly describe your algorithm and give its worst case cost.

(b) (7 points) You are building an Internet system, which requires computational maintenance every time the number of users is an increasing power of 2. For example, maintenance first occurs when the number of users is 2, then occurs next when this value is 4, then 8, and so forth. The computational cost of a user joining the system is 1, and the computational cost of maintenance is n, where n is the number of users in the system when maintenance occurs.Assume that users only join the system. What is the amortized cost per join? Show that this is the correct cost using the accounting method (3 sentences max to show that your taxes on join can pay for all costs).

(c) (7 points) Now users can join and leave the system, and the maintenance occurs every time the number of joins plus leaves is an increasing power of 2. For example, maintenance first occurs when the number of joins plus leaves is 2, then occurs next when this value is 4, then 8, and so forth. Computational cost is 1 for leaves, and remains the same for joins and maintenance. Now what is the amortized cost per join and leave? Show that this is the correct cost using the accounting method (4 sentences max to show that your taxes on join and leave can pay for all costs).

5. **Bob and His Coin**

(20 points) Suppose Bob keeps flipping a fair coin and each time scores one point for a head and two points for a tail. For a fixed $n \geq 1$, what is the probability that his score is precisely n points at some point? Hint: let $f(n)$ be the probability this happens, find a recurrence relation, and try to solve.