University of New Mexico
Department of Computer Science

# Midterm Examination

CS 561 Data Structures and Algorithms
Fall, 2017

| 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 - Asymptotics and Recurrences**

   Answer the following questions using the simplest possible *theta* notation. Assume as usual, that $f(n)$ is $\theta(1)$ for constant values of $n$. Briefly justify your answer where appropriate.

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

   (b) Solution to the following recurrence relation: $f(n) = 3f(n/2) + n$.

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

   (d) $\sum_{i=1}^{\log n} 2^i$

   (e) Minimum height of a binary tree with $n!$ leaf nodes?

2. **Short Answer - Probability, Algorithms and Data Structures** Answer the following questions using the simplest possible $\theta$ notation. Briefly justify your answer where appropriate.

   (a) Assume you have a graph with $n$ nodes and $m$ edge. Assume further that you have a new set-union data structure which somehow ensures all operations are $O(1)$ amortized time. What is the new runtime of Kruskal's algorithm?

   (b) Runtime of fastest algorithm to solve the Fractional Knapsack problem over $n$ items when the value per pound of each item is distributed independently and uniformly at random between 0 and 1.

   (c) In a skip list with $n$ items, each item has height exceeding $3 \log n$ with probability at most $1/n^3$. What is an upperbound on the probability that any item has height exceeding $3 \log n$?

   (d) Imagine you have two skip lists, each containing $n$ distinct items. What is the expected time to merge them into a new skip list over all $2n$ of the distinct items? The new skip list should be able to support all skip-list operations with the same time costs as for any skip lists over $2n$ items.

   (e) Alice tries to send a message to Bob over $n$ time steps. Assume Alice selects $\sqrt{n}$ time steps uniformly at random in which to send, and Bob selects $\sqrt{n}$ time steps uniformly at random in which to listen. What is the expected number of time steps in which both Alice sends and Bob listens?

3. **Dynamic Programming**

Consider the following variant of the donut buying problem from homework. Donuts are sold in boxes of two different quantities, $x_1$ and $x_2$ (where $x_1 < x_2$. We want to obtain exactly $n$ donuts if possible.

(a) (5 points) Assume that we want to obtain the minimum number of boxes. For example if the box quantities are $3, 4$ and $n = 16$, then the min number of boxes is 4 (four boxes of 4). For any positive $n$, let $m(n)$ be the minimum number of boxes needed to buy $n$ donuts if this is possible, or $+\infty$ otherwise. Write a recurrence relation for the value of $m(n)$. Don't forget the base case(s)!

(b) (12 points) Now assume that boxes have costs, $c_1$ and $c_2$, and we want to minimize our total cost to get $n$ donuts. For any positive $n$, now let $m(n)$ be the minimum cost needed to buy $n$ donuts if this is possible, or $+\infty$ otherwise. Write a recurrence relation for the value of $m(n)$. Don't forget the base case(s)!

(c) (3 points) Give an efficient algorithm for solving the donut buying with costs. How does its running time depend on $x_1$, $x_2$ and $n$? Is it an algorithm that runs in polynomial time in the input sizes?

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. The purges occur whenever the system size increases by 1/3. 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. **Coin Flips**

(a) (15 points) If you flip a fair coin $n$ times, what is the probability that you get a sequence that does not contain two heads in a row? Hint: Write a recurrence relation for the number of sequences of length $n$ that don't have two heads in a row; then use the solution to this recurrence to find the probability.

(b) (5 points) Let $p_n$ be the probability you computed in the part (a) - that a string of length $n$ does not have two heads. Using the values of $p_n$, write an expression for the expected number of coin flips before obtaining two heads. Hint: Use linearity of expectation.