University of New Mexico
Department of Computer Science

Midterm Examination

CS 362 Data Structures and Algorithms
Spring, 2021

Directions:

- This exam lasts 4 hours. It is open book, notes and Internet, but you are not allowed to discuss problems with any person.
- PLEASE: Start each main problem (i.e. Problems 1-5) at the top of a new page!
- Give yourself extra time to properly upload your solutions. Late exams may not be accepted.
- *Show your work!* You will not get full credit if we cannot figure out how you arrived at your answer.

1. **Short Answer (2 points each)**

   Answer the following questions using the simplest possible Θ notation unless otherwise stated. Assume in recurrences that $f(n)$ is $\Theta(1)$ for constant values of n.

   (a) What is $\sum_{i=1}^{n} i$?

   (b) What is $\sum_{i=1}^{n} \log i$?

   (c) Solution to the following recurrence $T(n) = 3T(n/3) + n$

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

   (e) Solution to the following recurrence relation: $f(n) = 3f(n-1) - 2f(n-2) + 2^n$. For this problem, give your answer in big-O notation.

(f) Solution to the following recurrence relation: $T(n) = 8T(\sqrt{n}) + \log^2 n$

(g) Worst-case runtime of randomized QuickSort on a list with n elements?

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

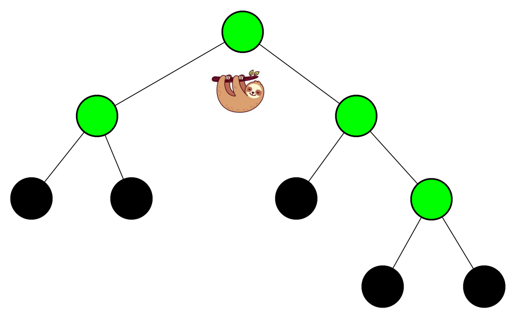
(i) Consider a certain operation OP. Say that over n calls to OP, the i-th call takes $\Theta(i)$ time. What is the amortized cost of OP?

(j) You flip n fair coins, and add each coin, one after the other, to the end of a row after flipping. Call a coin *satisfied* if it has the same value as its right neighbor in the row (the last coin in the row is never satisfied). What is the expected number of satisfied coins? For this problem, Give an exact answer (i.e. not just within Θ)

2. **Induction.** Deep in the Amazon lives a remarkable tree that grows downward from the forest canopy. This upside-down tree, beloved by both sloths and computer-scientists, is called an *s*-tree. Mathematically, an s-tree has green and black nodes obeying the following rules.

   - A green node has 2 children.
   - A black node has no children.

   Example *s*-tree:



   (a) (10 points) Prove, by induction on the number of nodes, n that an *s*-tree always has a number of black nodes that is 1 more than the number of green nodes. Don't forget to include the BC, IH, and IS.

(b) (10 points) Prove, by induction on the height, h that any s-tree has an odd number of nodes (recall that a number is odd if it can be written as $2x + 1$ for some integer x). Don't forget to include the BC, IH, and IS.

3. **Sloth Boxing** A bakery sells donuts in boxes of two different quantities, x_1 and x_2. Additionally, each box requires **work** w_1, w_2 to assemble. You are given an integer n and must return the minimum work set of boxes to get exactly n donuts, if n is possible; or ∞ if it is impossible.

   For example if $x_1 = 6$, $x_2 = 9$, $w_1 = 1$, $w_2 = 3$ and $n = 18$, then you should return: 3 boxes of size $x_1 = 6$, for total work of 3. However, if $n = 11$, you should return ∞.

   (a) (4 points) A coworker suggests the following greedy algorithm for this problem. Always use the box requiring minimum work. Give an example with x_i, w_i and n values, showing this greedy algorithm fails.

   (b) (12 points) For any positive x, let $m(x)$ be the minimum work set of boxes needed to get x donuts if this is possible, or ∞ otherwise. Write a recurrence relation for the value of $m(x)$. Don't forget the base case(s)!

   (c) (4 points) Give an efficient algorithm for solving Sloth Boxing. How does its running time depend on x_1, x_2 and n?

4. **Probability and Expectation**

(a) (4 points) A population of n sloths consists of 1 3-toed sloth, and $n-1$ 2-toed sloths. Every day, you catch and release a sloth selected independently and uniformly at random from the population. If you do this for n days, what is the expected number of times you catch the 3-toed sloth?

(b) (2 points) Using Markov's inequality, bound the probability that you catch the 3-toed sloth 10 times over n days.

(c) (4 points) Using a union bound, bound the probability that you ever catch the 3-toed sloth 2 days in a row over n days. *Hint: Compute the probability that you catch the* 3*-toed sloth on day* i *and day* $i+1$ *for any* i*. Then use a union bound to sum over values of* i*.*

(d) (10 points) Now, every day you catch, tag, and release a sloth selected independently and uniformly at random from the population of n sloths. If you do this for n days, what is the expected number of tagged sloths? *Hint: Define an indicator random variable for each of the* n *sloths which is* 1 *if the sloth is ever caught and* 0 *otherwise. What is the probability that a fixed sloth is not caught over* n *days? Next, use linearity of expectation!*

5. **Sloth Sequence (HARD!)** A sloth has a sequence S of numbers giving the calories obtainable by hunting on each day. The sloth must obtain at least T calories total, but wants to find a smallest sequence of contiguous days to do this. Formally, you are given a sequence $S = s_1, \ldots, s_n$ of numbers, and a number T. You want to find indices f and ℓ, $1 \leq f < \ell \leq n$ such that $\sum_{x=f}^{\ell} S[x] \geq T$, and $\ell - f$ is as small as possible. If the numbers in the sequence sum to less than T, then you will return NIL.

For example if $T = 8$, and $S = 2, 4, 1, 6, 0, 4, 5, 2, 3, 4$, then the answer should be the sequence $4, 5$ ($f = 6$, $\ell = 7$) since this gives the shortest contiguous sequence with value at least T.

You want to solve this problem in $o(n^2)$ time. You will use 2 functions, defined as follows:

- $lastIndex(i)$ returns the smallest index such that $\sum_{j=lastIndex(i)}^{i} s_j \geq T$, when $\sum_{j=1}^{i} s_j \geq T$. Returns the value 1 when $\sum_{j=1}^{i} s_j < T$.
- $sum(i)$ returns $\sum_{j=lastIndex(i)}^{i} s_j$

(a) (5 points) Fill in the remaining values for LastIndex() and sum() for the example sequence with $T = 8$ in the table below.

S	2	4	1	6	0	4	5	2	3	4
lastIndex	1	1	1	2	2					
sum	2	6	7	11	11					

(b) (10 points) Write recurrence relations for lastIndex(i) and sum(i) that can be calculated efficiently to fill in a table like the one above.

(c) (5 points) In $2-3$ sentences describe an efficient dynamic program that uses your recurrences to solve the Sloth Sequence problem. What is the run-time of your dynamic program?