University of New Mexico
Department of Computer Science

Midterm Examination

CS 561 Data Structures and Algorithms
Fall, 2021

Directions:

- 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.

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

1. **Short Answer (4 points each)**

   Answer the following questions using the simplest possible Θ notation.

   (a) Expected runtime of Bucket Sort on a list with n numbers independently and uniformly distributed between 0 and 1?

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

   (c) Solution to the recurrence: $T(n) = 4T(n/2) + n$

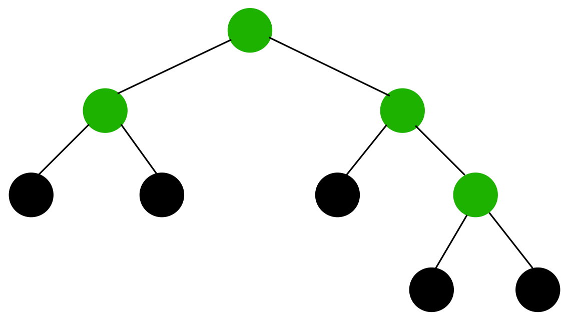
   (d) Solution to the recurrence: $f(n) = 5f(n-1) - 6f(n-2)$ (answer in big-O)

   (e) Consider a certain operation OP. Over n calls to OP, the i-th call, for $1 \leq i \leq n$, takes $\Theta(i)$ time. What is the amortized cost of OP?

2. **Induction.** An c-tree has green and black nodes obeying the following:

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

Example s-tree:



(a) (20 points) Prove, by induction on the number of nodes, n, that a c-tree always has a number of black nodes that is 1 more than the number of green nodes. *Hint: In the IS, let* g_1, b_1 *be the number of green and black nodes in left subtree of the root, and let* g_2, b_2 *be the number of green and black nodes in the right subtree.*

3. **Unboxed Donuts**

You have 3 types of boxes that fit quantities of x_1, x_2 and x_3 of donuts respectively. Your boss insists that any time a box is used, it must be completely filled with donuts. Your goal is to finding a boxing of donuts that minimizes the number of unboxed donuts.

(a) (12 points) Let $f(n)$ be the minimum number of unboxed donuts, when boxing n donuts. Write a recurrence relation for $f(n)$.

(b) (4 points points) Briefly describe a dynamic program based on this recurrence relation. Analyze the runtime.

(c) (4 points) Will a greedy algorithm that always puts remaining donuts in the largest box possible solve this problem? Give a proof or counter-example.

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. Consider the bank robbing problem, but where you now want to rob the bank on the smallest number of contiguous days. 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 must solve this problem in $o(n^2)$ time. You will use 2 functions, defined as follows:

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

   (a) (4 points) Fill in the remaining values for index() and sum() for the example sequence with $T = 8$ in the table below.

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

   (b) (8 points) Write recurrence relations for index(i) and sum(i) that can be calculated efficiently to fill in a table like the one above.

(c) (8 points) Describe an efficient dynamic program using your recurrences to solve this problem. Analyze the runtime of your algorithm. *Hint: In your analysis, think about the total work needed to fill in* all *cells of the table in the worst case. You may find ideas we discussed about amortized analysis helpful.*