CS3000: Algorithms & Data — Summer 2025 — Laney Strange

Homework 4
Due Thursday June 5th@ 9pm via Gradescope

Name:
Collaborators:

- Put your name on the first page. If you are using the LaTeX template we provided, then you can make sure it appears by filling in the `yourname` command.
- This assignment is due Thursday June 5th@ 9pm via Gradescope. You may submit up to 48 hours late for no penalty, but expect a delay in grading.
- Show ALL your work, even if the problem doesn't specify it.
- You'll have an opportunity to resubmit one homework at the end of the semester.
- Solutions must be typeset, preferably in LaTeX. If you need to draw any diagrams, you may draw them by hand as long as they are embedded in the PDF. I recommend using the source file for this assignment to get started.
- Any solutions that include pseudocode must be in the CLRS style.
- I encourage you to work with your classmates on the homework problems. *If you do collaborate, you must write all solutions by yourself, in your own words.* Do not submit anything you cannot explain. Please list all your collaborators in your solution for each problem by filling in the `yourcollaborators` command.
- If you get stuck on a homework problem, come by office hours or post on Piazza! We recommend you spend about 30 minutes trying to figure out a problem, and then ask for help. We'll be happy to clarify material from class and algorithm concepts, but we will not give out solutions or confirm your answers are correct.
- Finding solutions to homework problems online, or by speaking with students not enrolled in the class, is strictly forbidden.

Problem 1. *Hash Tables (5 points)*

Suppose you have a hash table that uses linked-list chaining to handle collisions. Normally we use unsorted, doubly-linked lists for this implementation; but what if we maintain sorted order for those lists?

(a) Describe how the find operation would work and give a bound on its worst-case run-time.

Solution:

(b) Describe how insert into the hash table would work and give a bound on its worst-case run-time.

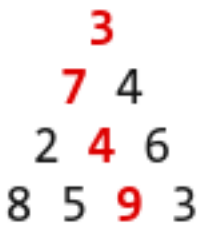
Solution:

(c) Describe how removal from a hash table would work and give a bound on its worst-case run-time.

Solution:

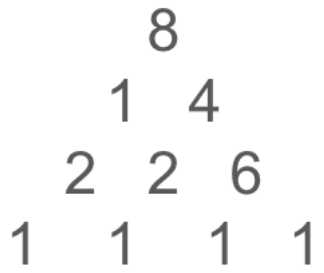
Problem 2. *Dynamic Programming (10 points)*

You are given a triangle of non-negative numbers with i numbers at row i. You want to find the maximum path sum starting at the top of the triangle and moving to any of the numbers on the bottom row. At each step you can go to any of the (up to) two neighboring numbers in the row below.



(See this link for an example: https://projecteuler.net/problem=67.)

(a) For the triangle below, what would be the value of an optimal solution?



Solution:

(b) Give a recursive equation $OPT(i, j)$ that would compute the value of the maximum path for row i up through position j.

Solution:

(c) Give complete pseudocode for a dynamic-programming solution to this algorithm and give a bound on its run-time.

Solution:

Problem 3. *Tatte Like Latte Part One (10 points)*

You have $\$D$ to spend on pastries at Tatte. In the bakery display, you see exactly one of each item – each has a price listed next to it, and you've personally assigned a rating of 1-10 as well. You want to spend your money in an optimal way, i.e., you want to maximize the sum of ratings on your items without going over $\$D$.

Here are the items you can buy, along with their prices and your individual ratings:

Item	Price	Your Rating
Chocolate Snail	\$5	9
Chocolate Croissant	\$4	7
Palmier	\$2	5
Monkey Bread	\$8	10

(a) What would an *optimal* solution be if you have \$10 to spend? What is the value of that solution (i.e., what is the sum of all the ratings)?

Solution:

(b) Going by ratings (largest to smallest), what would a Greedy solution be assuming you have \$10 to spend? Is it an an optimal solution?

Solution:

(c) In some versions of this problem, we compute the ratio of value (ranking) to weight (price), as shown in the table below. Using the rating-per-dollar as the way each item is evaluated, what would a Greedy solution be assuming you have \$10 to spend? Is it an an optimal solution?

Item	Price	Your Rating	Rating-Per-Dollar
Chocolate Snail	\$5	9	1.8
Chocolate Croissant	\$4	7	1.75
Palmier	\$2	5	2.5
Monkey Bread	\$8	10	1.25

Solution:

Problem 4. *Tatte Like Latte Part Two (10 points)*

You've eaten your fill of pastries, and now you have a new dollar amount to spend on Tatte's ground coffee beans. Like before, you've assigned a rating to each type of bean. Each type also has a price-per-pound listed, but you don't need to buy an entire pound of beans; instead, you can buy up to one pound of any type. You can also buy multiple types, as long as you don't go over your $\$D$ limit.

For example, here is what the menu might look like now.

Coffee Type	Price (per pound)	Your Rating
Arabica	\$8	5
Liberica	\$10	7
Excelser	\$12	8
Robusta	\$10	10

(a) Going by ratings (largest to smallest), what would a Greedy solution be assuming you have \$15 to spend and can purchase up to one pound of any coffee flavor? What would the total rating be?

Solution:

(b) In some versions of this problem, we compute the ratio of value (ranking) to weight (price), as shown in the table below. Using the rating-per-dollar as the way each item is evaluated, what would a Greedy solution be assuming you have \$15 to spend? What would the total rating be?

Item	Price (per pound)	Your Rating	Rating-Per-Dollar
Arabica	\$8	5	.625
Liberica	\$10	7	.7
Excelser	\$12	8	.667
Robusta	\$10	10	1

Solution:

(c) Give the pseudocode for a greedy solution that would go by either Rating, or Rating-per-Dollar, whichever is better based on your solutions above. (You can assume you have access to any sorting algorithm we've covered.) Your algorithm should take as parameters: D, the total dollar amount you can spend, an array of coffee prices, and a corresponding array of coffee ratings. It should return the total rating of an optimal solution.

Solution:

Problem 5. *Greedy == Optimal? (10 points)*

In one of the two Tatte problems, you found that the greedy strategy did well; your greedy choice was either the rating of an item, or rating-per-dollar. Now your job is to show that the greedy choice you made does yield an optimal solution.

(a) In which problem above did you find that Greedy gave you a pretty good solution? Was your greedy choice rating, or rating-per-dollar?

Solution:

(b) Show that your greedy choice would always yield an optimal solution for that problem. Start with defining S_k to be a subproblem of the Tatte problem, and let A_k be an optimal solution to S_k.

Solution: