CS3000: Algorithms & Data — Summer 2025 — Laney Strange

Homework 1
Due Thursday May 15 @ 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 May 15 @ 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. *Run-Time (5 points)*

(a) Express the function $n^3/1000 + 100n^2 - 100n + 3$ in terms of Θ notation.

Solution:

(b) List the following functions in increasing order of growth (slowest-growing to fastest-growing). As usual, lg means $\log_2$:

(a) 5^{5^n}

(b) $5n$

(c) 5^{5n}

(d) $4 \lg n$

(e) $2 \lg \lg n$

(f) n^6

(g) $5(n \lg n)$

(h) 5^n

(i) 5^{n^5}

(j) $(n/6)^6$

Solution:

Problem 2. *Mystery Algorithm (10 points)*

You encounter the following pseudocode, and you can assume it calls the procedure LINEARSEARCH implemented as we covered in class. The algorithm accepts an array of n distinct integers in the range $1..n+1$.

```
MYSTERY(A, n)
1  for i = 1 to n + 1
2      if LINEARSEARCH(A, n, i) == NIL
3          return i
```

(a) In English, describe what this algorithm is trying to accomplish.

Solution:

(b) Formally show the worst-case run-time $T(n)$ of the Mystery Algorithm. Assume that each execution of the kth line takes c_k time where c_k is a constant, and assume that LINEARSEARCH itself takes n steps.

Start by filling in the two rightmost columns of the table below.

line	cost	times
1		
2		
3		

Solution:

(c) Compute $T(n)$ based on your table above.

Solution:

(d) Give a tight bound on the run-time of this algorithm in the worst-case.

Solution:

(e) Give an example of A that would result in the best-case run time. (Your A example must be non-empty.) Give a bound on that run-time.

Solution:

Problem 3. *Improving Run-Time (10 points)*

Still assuming that the mystery algorithm above accepts an array of distinct integers in the range $1..n+1$.

(a) Give complete pseudocode, in the CLRS style, for an algorithm that will always return the same result, but with a more efficient run-time in the worst case. (But, sorting the array first is not an acceptable solution!).

Solution:

(b) Compute the worst-case run-time $T(n)$ of your algorithm, showing all your steps.

Solution:

(c) Give a tight bound on the worst case run-time.

Solution:

Problem 4. *Algorithmic Correctness (10 points)*

It's here again... our algorithm for LINEARSEARCH.

```
LINEARSEARCH(A, n, key)
1  for i = 1 to n
2      if A[i] == key
3          return i
4  return NIL
```

Prove that LINEARSEARCH is correct by showing that the following loop invariant: At the start of each iteration of the for loop, the subarray $A[1..i-1]$ consists of the elements in A that are not equal to the key. Use the loop invariant proof structure of (1) initialization, (2) maintenance, and (3) termination.

Solution:

Problem 5. *Recursive Sequences (10 points)*

(a) Find the four terms $a_2, ..., a_5$ of the sequence defined by the following recurrence relations.

- $a_n = 6a_{n-1}$. Base case: $a_1 = 2$
  Solution:

- $a_n = a_{n-1}^2$. Base case: $a_1 = 2$.
  Solution:

- $a_n = a_{n-1} + 3a_{n-2}$. Base cases: $a_0 = 1, a_1 = 2$.

  Solution:

(b) Use the iteration method to construct a closed-form formula for an arbitrary term a_k for the first recursive sequence above.

Solution:

(c) Give a recursive formula, including the base case, for the closed-form formula $a_k = \frac{(k)(k+1)}{2}$.

Solution: