CS3000: Algorithms & Data — Summer 2023 — Laney Strange

Recitation 2
Due May 16th @ 9pm via Gradescope

Name: Laney Strange
Collaborators: Huey, Duey, and Louie

- One recitation problem each week is graded; the rest are there for practice. That problem will be graded on completeness – full credit for making an honest effort. It is also closely linked to the quicksort problem on the upcoming Short Homework, so it's good to review it now!
- Recitations can be written by hand; submit a picture/screenshot on Gradescope.
- 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 recitation is due May 16th @ 9pm via Gradescope. If you miss the in-person recitation, or need to submit your solution later than the end of your section, please fill out this form: https://forms.gle/CLrhrkVauXYzC7U57
- Collaboration is strongly encouraged during recitation! Please list all your collaborators in your solution for each problem by filling in the `yourcollaborators` command.

Problem 1. *Quicksort (graded)*

(a) You can find the pseudocode for the Quicksort algorithm we did in class at https://course.ccs.neu.edu/cs3000/resources/Quicksort_Pseudocode.pdf. What is the run-time of the Partition step of Quicksort?

Solution:

(b) Give an example of an array that would result in the worst-case run-time of Quicksort.

Solution:

(c) Give a recurrence for the worst-case run-time of Quicksort.

Solution:

(d) Give an example of an array that would result in the best-case run-time of Quicksort.

Solution:

(e) Give a recurrence for the best-case run-time of Quicksort

Solution:

Problem 2. *Karatsuba (for practice; not graded)*

You can find the pseudocode we did in class for the Karatsuba algorithm at https://course.ccs.neu.edu/cs3000/resources/Karatsuba_Pseudocode.pdf

Let's walk through how the algorithm works when multiplying $7832 \cdot 3671$.

(a) The first call to Karatsuba is $\textsc{Karatsuba}(7832, 3671, 4)$. What are m, a, b, c, d?

Solution:

(b) The second call to Karatsuba is $ac = \textsc{Karatsuba}(78, 36, 2)$. What are m, a, b, c, d?

Solution:

(c) In the middle of that second call, we make three more calls to Karatsuba, but they will all end up in the base case. What are $ac, bd, bacd$ after returning from the base case calls?

Solution:

(d) Using those results, what does the second call return?

Solution:

(e) After all the recursive calls have completed, we're back in the first call to Karatsuba, like in Part A above. After all the recursive calls have completed, we now have $ac = 2808, bd = 2272, bacd = 1610$. What is the final answer?

Solution:

Problem 3. *Mergesort (for practice; not graded)*

Suppose you start with the following array: $< 8, 5, -9, 14, 0, -1, -7, 3 >$

(a) How many subarrays would mergesort split this array into in its first step? What would they be?

Solution:

(b) How many subarrays would mergesort split this array into in its second step? What would they be?

Solution:

(c) How many subarrays would mergesort split this array into in its third step? What would they be?

Solution:

(d) How many subarrays would exist after the first merge step? What would they be?

Solution:

(e) How many subarrays would exist after the second merge step? What would they be?

Solution:

(f) How many subarrays would exist after the third merge step? What would they be?

Solution:

(g) We know that the bound on Mergesort's worst-case run-time is $\Theta(n \lg n)$. What is the bound on its best-case run time?

Solution:

(h) Suppose you start with an already-sorted array. Describe a small change you can make to the Mergesort algorithm that could take advantage of this for a better real-world runtime.

Solution: