Assignment 2: More Sorting, Heaps
CS 758/858, Fall 2024
Due at **11:30pm on Sat Sep 7**

Implementation

Augment the code from Assignment 1 to implement quicksort (`quick`). For quicksort, you should use randomized partitioning (CLRS section 7.3). Those in 858 should also implement a second version of quicksort (`quick2`) that uses median-of-three partitioning (as in exercise 7.4–6) and insertion sort (as in exercise 7.4–5).

Written Problems

1. Briefly summarize which parts of your program are working or not. Include transcripts or plots showing the successes or failures. Is there anything else that we should know when evaluating your implementation work?
2. What can you conclude about the performance of the various algorithm implementations from your experiments? Please include your plots in your submission (electronic and hardcopy).
3. Give the time complexity of each operation when performed on an array and when performed on a linked list:
    (a) accessing the ith element,
    (b) inserting an element at the end (assuming the array already has unused space at the end),
    (c) inserting an element at the beginning (assuming the array already has unused space at the end),
    (d) constructing a copy with the same elements but in reverse order,
    (e) constructing a copy with the same elements in the same order but all occurrences of a given element x removed, and
    (f) modifying the original to remove all occurrences of a given element x.
4. Exercise 2.3–8 from CLRS.
5. Exercise 6.1–3 from CLRS.
6. Exercise 6.1–4 from CLRS.
7. Exercise 6.5-11 from CLRS.
8. (858 only) Exercise 7.2–2 from CLRS.
9. (858 only) Part b of problem 6–1 from CLRS
10. What suggestions do you have for improving this assignment in the future?

Submission

Please make sure that your code (as submitted) compiles on agate with the makefile that you supply. Please make sure that your code runs with the harness because this will be used to grade your assignment.

Electronically submit your source code as described in the instructions from the programming tips sheet on the course web page. Use the assignment name `2-undergrad` or `2-grad`, depending on whether you are enrolled in 858. Your files should include your `written.pdf` as well (feel free to use LaTeX or to scan/photograph handwritten work) and plots showing the performance of your implementation.

Evaluation

In addition to correctness, your work will be evaluated on clarity and efficiency.

Tentative breakdown: 2 mergesort, 2 quicksort, 6 written problems