

Assignment 3

SFWRENG 2CO3: Data Structures and Algorithms–Winter 2024

Deadline: February 4, 2024

Department of Computing and Software
McMaster University

Please read the *Course Outline* for the general policies related to assignments.

**Plagiarism is a serious academic offense and will be handled accordingly.
All suspicions will be reported to the Office of Academic Integrity
(in accordance with the Academic Integrity Policy).**

This assignment is an *individual* assignment: do not submit work of others. All parts of your submission *must* be your own work and be based on your own ideas and conclusions. Only *discuss or share* any parts of your submissions with your TA or instructor. You are *responsible for protecting* your work: you are strongly advised to password-protect and lock your electronic devices (e.g., laptop) and to not share your logins with partners or friends!

If you *submit* work, then you are certifying that you are aware of the *Plagiarism and Academic Dishonesty* policy of this course outlined in this section, that you are aware of the **Academic Integrity Policy**, and that you have completed the submitted work entirely yourself. Furthermore, by submitting work, you agree to automated and manual plagiarism checking of all submitted work.

Late submission policy. Late submissions will receive a late penalty of 20% on the score per day late (with a five hour grace period on the first day, e.g., to deal with technical issues) and submissions five days (or more) past the due date are not accepted. In case of technical issues while submitting, contact the instructor *before* the deadline.

Problem 1. Consider the following program

Algorithm SORT($L[0 \dots N]$):

Pre: L is an *array*.

- 1: **while** L is not sorted **do**
- 2: $L :=$ a random permutation of L .
- 3: **end while**

Post: L is sorted.

Assume we can test that L is sorted in $\Theta(|L|)$ and that we can compute a random permutation of L in $\Theta(|L|)$.

- P1.1. Does the SORT program sort correctly? If yes, then provide an invariant for the while-loop and provide a bound function that can be used to prove the correctness of the program. If no, then argue why the program is not correct.
- P1.2. Assume the program SORT is correct. Is the program stable? Explain why.
- P1.3. What is the worst case runtime complexity of this program? What is the best case runtime complexity of this program? Is this program optimal? Explain your arguments.
- P1.4. What is the expected case runtime complexity of this program? Explain your answer.

HINT: A *Bernoulli trial* is a random experiment with two possible outcomes: success and failure. If a Bernoulli trial T has a probability p of success, then the expected number of trials T needed

to get one success is $\frac{1}{p}$. For example, if you want to throw a six-sided dice until you see the face-value 4, then the probability of success is $\frac{1}{6}$ and the expected number of throws is 6.

Problem 2. The median of a list L of distinct values is the *middle* value $v \in L$: an equal number of values in L are smaller and larger than v . For example, in the list $L = [1, 5, 4, 2, 3]$, the median is 3.

Consider two sorted lists $A[0 \dots N]$ and $B[0 \dots M]$ with $N + M$ *distinct* values. You may assume that the total number of values in A and B is odd ($N + M$ is odd). Hence, there is a value $v \in (A \cup B)$ such that an equal amount $E = \lfloor \frac{N+M}{2} \rfloor$ of other values smaller and larger than v .

P2.1. Provide an algorithm that computes the *median* of the values in $A \cup B$ in $\Theta(\log_2(N + M))$.

HINT: Assume the median is at position p , $0 \leq p < N$, in A . How many values in A and B are smaller than the pivot? How many are larger?

P2.2. Explain why your algorithm is correct and why the complexity is $\Theta(\log_2(N + M))$.

P2.3. Let P be an algorithm with complexity $\Theta(\log_2(N + M))$ that computes the middle value in $A \cup B$. Argue how we can use P to break up the MERGE-step necessary to merge two sorted lists with $N + M = 2E + 1$ values into two independent MERGE-steps that each merge only E values.

Assignment Details

Write a report in which you solve each of the above problems. Your submission:

1. must start with your name, student number, and MacID;
2. must be a PDF file;
3. must have clearly labeled solutions to each of the stated problems;
4. must be clearly presented;
5. must *not* be hand-written: prepare your report in \LaTeX or in a word processor such as Microsoft Word (that can print or exported to PDF).

Submissions that do not follow the above requirements will get a grade of zero.

Grading

Each problem counts equally toward the final grade of this assignment.