Assignment 5

SFWRENG 2CO3: Data Structures and Algorithms–Winter 2024

Deadline: February 18, 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.

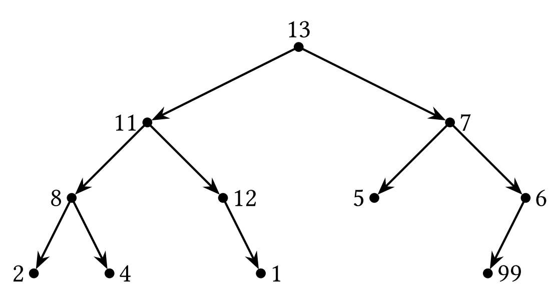
Problem 1. Consider the sequence of values $S = [3, 42, 39, 86, 49, 89, 99, 20, 88, 51, 64]$.

P1.1. Draw the min heap (as a tree) obtained by adding the values in S in sequence. Show each step.

P1.2. Draw the max heap (as a tree) obtained by adding the values in S in sequence. Show each step.

P1.3. Draw the binary search tree obtained by adding the values in S in sequence. Show each step.

Do *not* spend time drawing beautiful trees: a clear textual representation is good enough. Consider, for example, the following tree (note this tree is neither a heap or a binary search tree):



We can represent this tree textually via the representation:

```
13 (
    11 (
        8 (
            2
            4
        )
        12 (
            *
            1
        )
    )
    7 (
        5
        6 (
            99
            *
        )
    )
)
```

In this notation, we used `*` as a place holder for a missing child for those nodes that only have a single child.

Problem 2. Given an ordered list L and value v, the LowerBound algorithm provide the position p in list L such that p is the first offset in L of a value larger-equal to v. Hence, $v \leq L[p]$ (or, if no such offset exists, $p = |L|$). The LowerBound algorithm does so in $\Theta(\log_2(|L|))$ comparisons.

Argue that LowerBound is *worst-case optimal*: any algorithm that finds the correct position p for any inputs L and v using only comparisons will require $\Theta(\log_2(|L|))$ comparisons.

Problem 3. Min heaps and max heaps allow one to efficiently store values and efficiently look up and remove the *smallest values* and *largest values*, respectively. One cannot easily remove the largest value from a min heap or the smallest value from a max heap, however.

P3.1. Assume a value v is part of a min heap of at-most n values and that one knows v is stored at position p in that heap. Provide an algorithm that can remove v from the heap in worst-case $\mathcal{O}(\log_2(n))$.

P3.2. Provide a data structure that allows one to efficiently store values and efficiently look up and remove *both* the smallest and the largest values: all three of these operations should be supported in $\Theta(\log_2(n))$.

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 export 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.