Assignment 12

SFWRENG 2CO3: Data Structures and Algorithms–Winter 2024

Deadline: April 10, 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. For large integers, we cannot simply assume that we can compare them in constant times. Instead, large integers are typically represented as an array of digits and in the worst case, comparing two integers requires a per-digit comparison.

P1.1. Assume we have a list L of integers such that the *total* number of digits of all integers is D. Show how to sort L in $O(D)$.

P1.2. Assume we have an arbitrary list of N distinct integers. Show that the above algorithm will have a runtime complexity of at-least $\Theta(N \log_2(N))$ (without using the worst-case lower bound of at-least $\Omega(N \log_2(N))$ comparisons for general purpose comparison-based sorting).

Problem 2. Consider the following sentence:

"sally sells seashells by the seashore"

P2.1. Provide a table with the frequencies of each symbol in the above sentence (only consider the part between " and ").

P2.2. Use the HuffmanPFTrie algorithm to produce an optimal prefix-free code with respect to the frequencies of symbols in the above sentence (only consider the part between " and "). Show each step of the algorithm.

*This assignment is intended as a *bonus* and as an opportunity to improve ones grades if needed. Remember that the best ten out of twelve assignments will count toward your final grade. If you are happy with your previous assignments, then you are *not* required to make this assignment. The material covered in this assignment will *not* be part of the final exam.

P2.3. Encode the above sentence using the prefix-free code you determined for the above sentence.

P2.4. Assume we end up in an alternative universe in which computers do not use *bits* for as their smallest possible values, but instead uses *ternary digits (trits)*. We will represent the possible values for a trit by -1, 0, and 1. Having *three* elementary values allows us to produce shorter optimal prefix-free codes than one would get using only bits. Show how to adapt Huffman coding to provide optimal prefix-free codes using trits and argue that your approach is correct.

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.