Computer Science C73
Scarborough Campus

September 23, 2026
University of Toronto

Homework Assignment #2
Due: September 30, 2026, by 11:59 pm

- ***You must submit your assignment through the Crowdmark system.*** *You will receive by email an invitation through which you can submit your work in the form of separate PDF docucments with your answers to each question of the assignment. To work with a partner, you and your partner must form a group on Crowdmark.* ***The group must be formed before either partner submits an answer.***
- *To minimize bias in grading, Crowdmark does not reveal your name to the grader. Please do not subvert this feature by including your name(s) on the files you submit.*
- *The PDF files you submit must be legible without undue effort by the grader. To this end, I encourage you to learn and use the LaTex typesetting system, which is designed to produce high-quality documents that contain mathematical notation. You are not required to produce the PDF files you submit using LaTex; you may produce them any way you wish, as long as the resulting document is legible.*
- *By virtue of submitting this assignment you (and your partner, if you have one) acknowledge that you are aware of the policy on homework collaboration for this course.*[a]
- *For any question, you may use data structures and algorithms previously described in class, or in prerequisites of this course, without describing them. You may also use any result that we covered in class, or is in the course notes by referring to it.*
- *Unless we explicitly state otherwise, you may describe algorithms in high-level pseudocode or point-form English, whichever leads to a clearer and simpler description. Do not provide executable code.*
- *Unless we explicitly state otherwise, you should justify your answers. Your paper will be graded based on* ***the correctness and efficiency of your answers, and the clarity, precision, and conciseness of your presentation****.*

[a] "**For help with your homework you may consult only the instructor, TAs, your homework partner (if you have one), the CSCC73 Course Notes, and your personal class notes. You may not consult any other source, including AI-based tools.**"

Question 1. (10 marks) The CSCC73 midterm test has been graded. Each of the k TAs has sorted his or her pile of exams alphabetically. Each pile contains exactly n papers. The TAs have all gone home, and Vassos is stuck with merging these k sorted piles into a single pile of kn papers, sorted alphabetically.

a. Here is one algorithm that Vassos can use to solve his problem: Merge the first two sorted piles into a (sorted) pile; then merge the resulting pile with the third sorted pile; then merge the resulting pile with the fourth sorted pile, etc. What is the running time of this algorithm in terms of k and n? (Merging two sorted piles of papers can be done in time proportional to the size of the resulting pile.)

b. Give a more efficient divide-and-conquer algorithm that Vassos can use to solve this problem. What is the running time of your algorithm? (You don't need to justify its correctness.)

Question 2. (15 marks) An ***interval*** is a pair (ℓ, r) of numbers such that $\ell \leq r$, or the ***empty interval*** denoted $\varnothing$. The ***intersection*** of two intervals (ℓ, r) and (ℓ', r') is the interval $\big(\max(\ell, \ell'), \min(r, r')\big)$ if $\max(\ell, \ell') \leq \min(r, r')$; otherwise, it is $\varnothing$. The ***length*** of interval (ℓ, r) is $r - \ell$, and the length of $\varnothing$ is 0.

We are given a list of $n \geq 2$ nonempty intervals $(\ell_1, r_1), (\ell_2, r_2), \ldots, (\ell_n, r_n)$. We want to find two distinct intervals in this list whose intersection is as long as possible. It is straightforward to do this in $\Theta(n^2)$ time. Describe a divide-and-conquer algorithm that solves this problem in $O(n \log n)$ time. Explain why your algorithm is correct and analyze its running time.

Question 3. (10 marks) Let S be a set of at least two numbers, and let

$$\mathbf{ads}(S) = \frac{\max(S) - \min(S)}{|S| - 1}.$$

Intuitively, this is the average distance between pairs of successive elements of S in sorted order: there are $|S| - 1$ such pairs, and the distances between all of them add up to $\max(S) - \min(S)$. We say that $x, y \in S$ are ***close in*** S if $x \neq y$ and $|x - y| \leq \mathbf{ads}(S)$ — i.e., the distance between x and y is at most the average distance between successive elements of S. Note that any set of at least two numbers must contain a pair of close numbers: this is for the same reason that it is impossible for all of you to receive an above-average grade! Also note that a pair of numbers that are close in S need not be successive in sorted order; nor is every pair of successive numbers necessarily close in S.

Describe a divide-and-conquer algorithm that, given a set S of $n \geq 2$ numbers in arbitrary order, finds a pair of numbers in S that are close; your algorithm should run in $O(n)$ time. Justify the correctness and running time of your algorithm.

Hint. Use the following fact, which you should prove: Let p be any element of S, $S_0 = \{x \in S : x \leq p\}$ and $S_1 = \{x \in S : x \geq p\}$. (Note that p belongs to both S_0 and S_1, and so $|S_0| + |S_1| = |S| + 1$.) Let $a_i = \mathbf{ads}(S_i)$, for $i \in \{0, 1\}$, and let $m \in \{0, 1\}$ be such that $a_m = \min(a_0, a_1)$. Then *every* close pair in S_m is a close pair in S.