Computer Science C73 — September 9, 2019
Scarborough Campus — University of Toronto

Homework Assignment #1
(worth 4% of the course grade)
Due: September 16, 2019, by 10 am

• *You must submit your assignment as a PDF file through the MarkUs system by logging in with your UTORid at* `markus.utsc.utoronto.ca/cscc73f17`*. To work with a partner, you and your partner must form a group on MarkUs.*
• *It is your responsibility to ensure that the PDF file you submit is legible. 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 file you submit using LaTex; you may produce it 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 assigned sections of the official course textbooks, 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.*
• *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] "In each homework assignment you may collaborate with at most one other student who is currently taking CSCC73. If you collaborate with another student on an assignment, you and your partner must submit only one copy of your solution, with both of your names. The solution will be graded in the usual way and both partners will receive the same mark. Collaboration involving more than two students is not allowed. **For help with your homework you may consult only the instructor, TAs, your homework partner (if you have one), your textbook, and your class notes. You may not consult any other source.**"

Question 1. (20 marks) Let $\mathcal{I}$ be a sequence of closed intervals. A set of numbers C is a ***cover*** of $\mathcal{I}$ if every interval in $\mathcal{I}$ contains at least one number in C (that is, for each $I = [s, f]$ in $\mathcal{I}$, there is some $c \in C$ such that $s \leq c \leq f$). C is a ***minimum cover*** of $\mathcal{I}$ if it is a minimum cardinality cover of $\mathcal{I}$.

a. (5 marks) Consider the following greedy algorithm.

1 $C := \varnothing$
2 **while** $\mathcal{I} \neq \varnothing$ **do**
3 let c be a number that is contained in the largest number of intervals in $\mathcal{I}$
4 $C := C \cup \{c\}$
5 delete from $\mathcal{I}$ all intervals that contain c
6 **return** C

(In line 3, assume that ties are broken arbitrarily. That is, if there are multiple numbers that are contained in the same maximum number of intervals in $\mathcal{I}$, then we pick any one of these numbers as our c.)

Prove that this algorithm does not always find a minimum cover of $\mathcal{I}$. For full marks, your counterexample should be such that if in any iteration of the algorithm there are multiple choices for c, then ***every*** choice results in a suboptimal C at the end. (Thus, there is no hope of fixing this greedy algorithm by using a more refined choice of c as a number that is contained in the maximum number intervals ***and*** satisfies some additional property.)

b. (15 marks) Describe a greedy algorithm that given a sequence of intervals $\mathcal{I}$ (in arbitrary order) returns a minimum cover of $\mathcal{I}$. (Your algorithm should be given in clear high-level pseudocode or point-form English; do not present detailed code.) Prove the correctness of your algorithm.

Question 2. (10 marks) Consider the Interval Scheduling problem, where we are given a sequence $\mathcal{I}$ of closed intervals (in any order) and we want to find an optimal set of intervals in $\mathcal{I}$, i.e., a maximum cardinality set of non-overlapping intervals in $\mathcal{I}$. We have seen that the "earliest-finish-time-first" greedy algorithm finds an optimal set of intervals. We have also seen that the "fewest-overlaps-first" greedy algorithm, shown below, does ***not*** always find an optimal set of intervals.

FewestOverlapsFirst($\mathcal{I}$)
1 $A := \varnothing$
2 **while** $\mathcal{I} \neq \varnothing$ **do**
3   let I be an interval in $\mathcal{I}$ that overlaps the fewest intervals in $\mathcal{I}$
4     $A := A \cup \{I\}$
5   delete from $\mathcal{I}$ all intervals that overlap I
6 **return** A

(In line 3 ties are broken arbitrarily.) In this problem you must prove that FewestOverlapsFirst produces a set of non-overlapping intervals whose size is at least half the size of an optimal set. More precisely, let m be the size of an optimal set of intervals, and let A be the set of intervals computed by the FewestOverlapsFirst greedy algorithm. Prove that $|A| \geq m/2$.

This implies that, in some sense, the set of intervals produced by this algorithm is not too bad compared to the optimum. (Contrast this to the "earliest-start-time-first" greedy algorithm, which produces a set of intervals whose size can be an arbitrarily small fraction of the optimal — can you see why?)

Hint. Let B be any optimal set of intervals and I be any interval in A. What is the maximum number intervals in B with which I overlaps?