

Homework Assignment #1  
Due: September 13, 2023, 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 documents with your answers to each question of the assignment. To work with a partner, you and your partner must form a group on Crowdmark. Crowdmark does not enforce a limit on the size of groups. **The course policy that limits the size of each group to at most two remains in effect:** submissions by groups of more than two persons will not be graded.
- It is your responsibility to ensure that the PDF files you submit are 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 files 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.<sup>a</sup>
- 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. 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.**

<sup>a</sup>“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) (All intervals in this question are assumed to be closed real intervals with more than one element each, i.e., they are of the form  $[s, f]$ , where  $s, f \in \mathbb{R}$  and  $s < f$ .) A set of intervals  $\mathcal{J} = \{I_1, I_2, \dots, I_n\}$  **covers** an interval  $I$  if  $I \subseteq \cup_{i=1}^n I_i$ . We are given an interval  $I$  and a finite set of intervals  $\mathcal{J}$  that covers  $I$ . We wish to find a subset of  $\mathcal{J}$  that covers  $I$  and has as few intervals as possible.

(Think of  $I$  as the duration of some student event, and  $I_1, I_2, \dots, I_n$  as the periods of time during which each of  $n$  volunteers can staff the AMACSS booth during the event. We want to find the smallest number of volunteers so that at all times during the event at least one volunteer is staffing the AMACSS booth.)

**a.** Consider the following greedy strategy. Start with the empty set of intervals  $A$ ; in each iteration add to  $A$  the interval in  $\mathcal{J}$  that has the largest intersection with the uncovered part of  $I$ , i.e.,  $I - \cup_{J \in A} J$ . (Note that initially the uncovered part of  $I$  is all of  $I$ , since at that time  $A$  is empty.) Repeat this until all of  $I$  is covered (i.e., until  $I \subseteq \cup_{J \in A} J$ ). Give a counterexample showing that this strategy does not necessarily yield a desired subset of  $\mathcal{J}$ .

**b.** Describe a greedy algorithm to find a subset of  $\mathcal{J}$  that covers  $I$  and has as few intervals as possible. Prove that your algorithm is correct, and analyze its running time.

[continued on next page]

**Question 2.** (15 marks) Let  $T$  be a (rooted) tree, not necessarily binary, and  $A$  be a subset of nodes of  $T$ . Given  $T$  and  $A$ , we want to find a **maximum cardinality subset**  $B$  of  $A$  such that the subtrees rooted at the nodes of  $B$  are pairwise disjoint.

Describe a greedy algorithm that solves this problem. Prove that your algorithm is correct, and analyze its running time. For the purpose of the running time analysis, assume that the nodes of the tree are labeled with the positive integers 1 to  $n$ ; the root is the node labeled 1; and for each node  $i \in [1..n]$ , we are given (a) the list of its children,  $children(i)$  (empty, if  $i$  is a leaf), and (b) the parent of  $i$ ,  $parent(i)$  (0, if  $i = 1$ ).