CSC 545 **Midterm Exam** **Spring 2022**

Name _____

Instructions This exam consists of a single problem worth 100 points. From Problems (1) and (2), choose *one*. (If you do more than one problem, only one will be graded.)

When designing a *divide and conquer* algorithm, be sure to:

(a) describe your algorithm using prose and pictures,
(b) argue that your algorithm is correct, and
(c) analyze its running time.

Pseudocode is not required. When analyzing the time taken by a recursive algorithm, write down a recurrence and solve it.

When designing a *dynamic programming* algorithm, be sure to follow the four-part framework. You will be graded on each part of the framework, with the following weighting:

(1) solution structure, 30%;
(2) recurrence equation, 50%;
(3) evaluation phase, 10%;
(4) recovery phase, 10%.

During this take-home midterm exam, the only resources you are allowed to use are: your personal *course notes*, the *course handouts*, the *textbook*, and the videos of *recorded lectures* from this course. Individual work is required. Use of any resource in preparing your answers *must be cited*.

You must submit your answers on `Gradescope` (at `www.gradescope.com`) for the course CSC 545 by 11:59pm MST on Thursday, March 3. Upload a single PDF file containing your submission, and be sure to mark which pages of your submission correspond to which problem. If you write your answers by hand on paper, use an application to scan your pages (such as the smartphone apps `CamScanner` or `TurboScan`).

I will be available online through `Zoom` during 3:30–4:45pm MST to answer questions concerning the exam at:

`https://arizona.zoom.us/j/5128057331`

Please only enter the waiting room if you have a question about the exam.

Good luck!

(1) **(Finding the kth smallest in the merge of four arrays)** (100 points) Suppose we are given four *sorted* arrays $A[1:n]$, $B[1:n]$, $C[1:n]$, and $D[1:n]$ (where within each array their elements are in increasing order and the elements in all the arrays are distinct), together with an integer k where $1 \leq k \leq 4n$. We would like to identify which element in these four arrays would be the kth-smallest element in the *merge* of arrays A, B, C, and D into one sorted array.

Using *divide and conquer*, design an algorithm that, given the four separate sorted arrays A, B, C, and D as input, finds the kth-smallest element in their merge in $O(\log k)$ time. Be sure to argue that your algorithm is correct.

(Note: You cannot actually merge arrays A, B, C, and D, as that would take $\Theta(n)$ time, which exceeds the allowed time bound.)

(2) **(Scheduling a theater)** (100 points) Suppose you are the owner of a theater, and several different groups want to schedule shows in your theater. For each group, you know when they want their show to run, and you have an estimate of the amount of income you will make from ticket sales. Since you cannot schedule all the groups in your theater, you want to select a subset that can be scheduled and that will maximize your total estimated income from their shows.

Formally, this problem can be modeled as follows. You are given a collection of n tasks, each of which is represented by its time interval with real-valued endpoints,

$$[a_1, b_1),\ [a_2, b_2),\ \ldots,\ [a_n, b_n),$$

and its corresponding real-valued weight,

$$w_1,\ w_2,\ \ldots,\ w_n.$$

Task i uses interval $[a_i, b_i)$ and has weight w_i.

Using *dynamic programming*, design an algorithm that finds a subset S of the tasks, $S \subseteq \{1, \ldots, n\}$, such that the tasks in S are are *nonoverlapping*,

$$[a_i, b_i) \cap [a_j, b_j) = \emptyset \quad \text{for all} \quad i, j \in S \text{ where } i \neq j,$$

and that the tasks in S have *maximum total weight*,

$$\sum_{i \in S} w_i.$$

Your algorithm should run in $O(n^2)$ time.