Algorithms: CSE 202 — Mid-term I
Section A

Date: October 25, 2018
Maximum Points: 25
Time: 75 minutes

Name:
ID:

Solve all THREE problems. Problems carry points as indicated. In each case, present a high-level description of the algorithm along with any definitions and recursive formulations. Also present a brief correctness argument together with an analysis of the time complexity. Obtain as efficient an algorithm as possible.

Problem 1: Unit intervals with weights

Describe an efficient algorithm that, given a set $\{x_1, x_2, \ldots, x_n\}$ of points on the real line where the i-th point has weight $w_i \geq 0$ for $1 \leq i \leq n$, selects at most $k \geq 0$ unit-length closed intervals so that the sum of weights of the points they contain is maximized. Argue that your algorithm is correct. State its complexity.

A unit length interval is an interval of the form $[y, y+1]$ where y is any real number. A point x_i is contained in the unit interval $[y, y+1]$ if $y \leq x \leq y+1$.

It is important to note that one can select any unit interval $[y, y+1]$ that begins at a real number y. However, the one cannot select more than k such intervals.

Problem 2: Covering intervals with points

Describe an efficient algorithm that, given a set $\{[s_i, s_i+1] |\ 1 \leq i \leq n\}$ of closed unit intervals on the real line where the i-th interval has weight $w_i \geq 0$, selects $k \geq 0$ points on the real line, so that the sum of the weights of the intervals that are hit by the selected points is maximized.

A point x_j hits the interval $[s_i, s_i+1]$ if the point is inside the interval, that is, $s_i \leq x_j \leq s_i+1$. You can select any real number to hit intervals. However, you can only select k of them. If an interval is hit by more than one point, you can only add its weight once to the sum. The goal is to select k points so that the sum of the weights of the intervals hit by the selected points is maximized.

You can assume that all the s_i are distinct. Argue that your algorithm is correct. State its complexity.

Problem 3: Covering with weighted intervals

Describe an efficient algorithm that, given a set $\{x_1, x_2, \ldots, x_n\}$ of points on the real line and a set of closed intervals $[s_i, f_i]$ for $1 \leq i \leq d$ where the cost of the i-th interval is $c_i > 0$, determine the least costly set of closed intervals that contains all of the given points. Assume that there is a set of intervals to cover all the points.

Problem 4: Long trip wiht fixed days (DPV)

You are going on a long trip. You start on the road at mile post 0 Along the way there are n hotels at mile posts $0 < a_1 < a_2 < \cdots < a_n$, where each a_i is measured from the starting point. The only places you are allowed to stop are at these hotels, but you can choose which of the hotels you stop at. You must stop

at the final hotel (at distance a_n), which is your destination. Moreover, you are required to complete your journey in exactly d days.

You'd ideally like to travel the same distance every day, but this may not be possible (depending on the spacing of the hotels). If you travel x miles during a day, the *penalty* for that day is x^2. You want to plain your trip so as to minimize the total penalty - that is, the sum, over all travel days, of the daily penalties. Give an efficient algorithm that determines the optimal sequence of hotels at which to stop.

Problem 5: Unit intervals with square weights

Describe an efficient algorithm that, given a set $\{x_1, x_2, \ldots, x_n\}$ of points on the real line, determines the set of unit-length closed intervals with the smallest weight that contains all of the given points. Argue that your algorithm is correct. State its time complexity.

Assume that $x_i \geq 1$ for all $1 \leq i \leq n$. A unit interval is of the form $[y, y+1]$ where y is a real number. The weight of such an interval is y^2. A point x_i is contained in the unit interval $[y, y+1]$ if $y \leq x_i \leq y+1$. The weight of a set of unit intervals is the sum of the weights of the intervals in the set. For example, the weight of the set of intervals $\{[y_j, y_j+1] |\ 1 \leq i \leq k\}$ is $\sum_{j=1}^{k} y_j^2$. The goal is to select a set of unit intervals to cover all the points such that their sum of weights is minimized.

Problem 6: Product of sums maximization

Suppose you are given two sets A and B, each containing n positive integers. You can choose to reorder each set however you like. After reordering, let a_i be the ith element of set A, and let b_i be the ith element of B. You will receive a payoff of $\prod_{i=1}^{n}(a_i + b_i)$. Give an algorithm that will maximize your payoff. Prove that your algorithm maximizes the payoff, and state its running time.

Problem 7: Worker partnerships

Consider the following problem: After millions of years of warfare between the continents of Ur and Nena, the proletariat of the continents decided that it is best to work together to advance the civilization. Each continent has selected a set of n workers to team with the workers from the other continent. Each team will consist of two workers, one from each continent, so they can work together on a project for the benefit of both the continents. However, it is not an easy matter for workers from different continents to work together productively. Each worker has a degree of adaptability to work with a worker from the other continent. Let p_i for $1 \leq i \leq n$ be the degree of adaptability of the i-th worker from the continent Ur. Let q_i for $1 \leq i \leq n$ be the degree of adaptability for the i-th worker from the continent Nena. If worker i from Ur and worker j from Nena formed a pair, then the productivity of the pair is $p_i q_j$, the product of their degrees of adaptability. Our goal is to form as many pairs of workers as possible where one worker is chosen from each set so that for each pair the product of their degrees of adaptability is at least x.

Each worker can only be in at most one team. Each team has exactly two workers, one from each continent. We want to maximize the number of teams.

Problem 8: Cookies

Consider the following problem:

You are baby-sitting n children and have $m \geq n$ cookies to divide among them. You must give each child exactly one cookie (of course, you cannot give the same cookie to two different children). For $1 \leq i \leq n$, child i has a greed factor g_i, which is the minimum size of a cookie that the child will be content with; and for $1 \leq j \leq m$ cookie j has a size s_j. Your goal is to maximize the number of content children. Child i is content if it is assigned cookie j such that $c_i \leq s_j$.

Problem 9: Classes and rooms

You are given a list of classes C and a list of classrooms R. Each class c has a positive enrollment $E(c)$ and each room r has a positive integer capacity $S(r)$. You want to assign each class a room in a way that minimizes the total sizes (capacities) of rooms used. However, the capacity of the room assigned to a class must be at least the enrollment of the class. You cannot assign two classes to the same room. Design an efficient algorithm for assigning classes to rooms and prove the correctness of your algorithm.

Problem 10: Difference of two items

The input is an array A containing n real numbers, and a real number x.

1. Design an algorithm to determine whether there are two elements of A whose difference is exactly x. The algorithm should run in time $O(n \lg n)$.
2. Suppose now that the array A is given in a sorted order. Design an algorithm to solve this problem in time $O(n)$.

Problem 11: Two-product

The input is an array A containing n non-negative intergers, and an integer x.

1. Design an algorithm to determine whether there are two elements of A whose product is exactly x. The algorithm should run in time $O(n \lg n)$.
2. Suppose now that the array A is given in a sorted order. Design an algorithm to solve this problem in time $O(n)$.

Problem 12: Combine intervals

Given a set of time intervals in any order, merge all overlapping intervals into one. Output the result, which should have only mutually exclusive intervals. Let the intervals be represented as pairs of integers for simplicity. For example, the intervals $[1, 5]$, $[3, 8]$ and $[6, 10]$ should be merged into $[1, 10]$.

Problem 13: Finding the peak in a unimodal sequence

Let $a_i, \ldots, a_n$ be a sequence of distinct integers. Moreover, the sequence is unimodal, that is, there is an index $1 \leq p \leq n$ such that the values a_i increase up to position p and decrease from then on. Design an efficient algorithm to find the peak value.