

PROBLEM SET 4
Due: Monday, March 5

Important: *On Monday, March 5, you will receive a take-home exam. In order to give you maximal time to work on your exam, I require Problem Set 4 to be turned in no later than 11:59pm on Monday, March 5. Solutions to Problem Set 4 will be made available shortly thereafter.*

Reading: CLR Chapters 7, 9 & 10.

Suggested Problems: 7.1-5; 7.2-1; 7.3-1; 7.4-1 7.4-2; 7.5-3; 7.5-6; 9.1-1, 9.1-4; 9.2.3; 10.3-1, 10.3-3, 10.3-6, 10.3-7, 10.3-8, 10.3-9; 10-1

Note: This problem set is different in character from the previous problem sets because all of the problems ask you to design and describe algorithms. Your descriptions should be concise, but should give enough detail that anyone familiar with CLR would know how to implement the algorithm from your description. In particular, you should liberally use algorithms we have studied as black boxes. E.g., you might describe one step of an algorithm as “use CLR’s linear order statistic algorithm to find the 17th largest element of the array” or “use mergesort to sort the array in worst-case $O(n \lg n)$ time.” However, if you modify an algorithm we have studied to do something different, you must carefully specify the steps of your algorithm. Unless specifically requested, you do not have to write pseudocode for your algorithms, but you are encouraged to do so if you find that it’s the best way to explain yourself. If you are asked to design an algorithm with a particular running time, you must justify *why* your algorithm has that running time.

Problem 1 [20] Exercise 9.2-5 on p. 178 of CLR. Note that the notation $[a..b]$ means the range of integers from a to b , *inclusive*.

Problem 2 [20] Exercise 9.3-4 on p. 180 of CLR. *Hint:* View an integer x in the range $[1..n^2]$ as a pair of “digits” $(x \text{ div } n, x \text{ mod } n)$.

Problem 3 [25] Exercise 10.1-1 on p. 187 of CLR. *Hint:* Imagine that the comparisons induce a tree over the elements that represents a tournament in which each node has a winner (smaller element) and a loser (larger element). Be careful that your algorithm accounts for *all* comparisons performed.

Problem 4 [35] Do parts a, b, and c of Problem 10-2 on p. 193 of CLR. Do *not* do parts d and e. For part c, give pseudocode (using SELECT as a black box) and justify why your algorithm is $\Theta(n)$.

The notation in this problem is somewhat confusing. Here’s one way to view the problem. You are given an array A with n elements, each of which is a record with two real-numbered fields named x and w . What the problem calls x_i is $A[i].x$ and what it calls w_i is $A[i].w$. Thus, each value x has an associated weight w that does not change throughout the problem. All the x values are distinct, but the weights are not guaranteed to be distinct.

In the notation $\sum_{x_i < x_k} w_i \frac{1}{2}$, the sigma summation notation means "The sum of the weights of all x_i such that $x_i < x_k$ ". It does *not* mean "The sum of the weights of all x_i such that $i < k$ ". The other sigma sum is symmetric.

The weighted median of a collection is not necessarily unique. For example, if each of the four elements 1, 2, 17, and 72 has weight 0.25, then both 2 and 17 are weighted medians.

Grading:

- Part a: [5]
- Part b: [10]
- Part c: Algorithm [15]
Analysis [5]

Extra Credit Problem [20] Exercise 10.3-4 on p. 192 of CLR.

Problem Set Header Page
Please make this the first page of your hardcopy submission.

CS231 Problem Set 4
Due Monday, March 5, 2001

Name:

Date & Time Submitted (*only if late*):

Collaborators (*anyone you collaborated with in the process of doing the problem set*):

*In the **Time** column, please estimate the time you spent on the parts of this problem set. Please try to be as accurate as possible; this information will help me to design future problem sets. I will fill out the **Score** column when grading your problem set.*

Part	Time	Score
General Reading		
Problem 1 [20]		
Problem 2 [20]		
Problem 3 [25]		
Problem 4 [35]		
Extra Credit [20]		
Total		