

# CS 4349.400 Homework 3

Due Wednesday September 19th, in class

Please answer each of the following questions.

1. Suppose we are given a set  $S$  of  $n$  items, each with a *value* and a *weight*. For any element  $x \in S$ , we define two subsets

- $S_{<x}$  is the set of all elements of  $S$  whose value is smaller than the value of  $x$ .
- $S_{>x}$  is the set of all elements of  $S$  whose value is larger than the value of  $x$ .

For any subset  $R \subseteq S$ , let  $w(R)$  denote the sum of the weights of elements in  $R$ . The **weighted median** of  $R$  is any element  $x$  such that  $w(S_{<x}) \leq w(S)/2$  and  $w(S_{>x}) \leq w(S)/2$ .

Describe and analyze an algorithm to compute the weighted median of a given weighted set in  $O(n)$  time. Your input consists of two unsorted arrays  $S[1..n]$  and  $W[1..n]$ , where for each index  $i$ , the  $i$ th element has value  $S[i]$  and weight  $W[i]$ . You may assume that all values are distinct and all weights are positive.

2. In a previous life, you worked as a cashier in the lost Antartican colony of Nadira, spending the better part of your day giving change to your customers. Because paper is a very rare and valuable resource in Antarctica, cashiers were required by law to use the fewest bills possible whenever they gave change. Thanks to the numerological predilections of one of its founders, the currency of Nadira, called Dream Dollars, was available in the following denominations: \$1, \$4, \$7, \$13, \$28, \$52, \$91, \$365.<sup>1</sup>
  - (a) The greedy change algorithm repeatedly takes the largest bill that does not exceed the target amount. For example, to make \$122 using the greedy algorithm, we first take a \$91 bill, then a \$28 bill, and finally three \$1 bills. Give an example where this greedy algorithm uses more Dream Dollar bills than the minimum possible. [Hint: It may be easier to write a small program than to work this out by hand.]
  - (b) Describe a recursive algorithm that computes, given an integer  $k$ , the minimum number of bills needed to make  $k$  Dream Dollars. **Express your running time using a recurrence relation.** You *do not* need to solve the running time recurrence to get full credit. (And don't worry about making your algorithm fast; just make sure it's correct. We'll learn how to make it fast next week.)
3. A *subsequence* of a sequence  $A$  consists of a (not necessarily contiguous) collection of elements of  $A$ , arranged in the same order as they appear in  $A$ . If  $B$  is a subsequence of  $A$ , then  $A$  is a *supersequence* of  $B$ .

---

<sup>1</sup>For more details on the history and culture of Nadira, including images of the various denominations of Dream Dollars, see <http://moneyart.biz/dd/>.

- (a) Describe a simple *recursive* algorithm to compute, given two sequences  $A[1..m]$  and  $B[1..n]$ , the length of the *longest common subsequence* of  $A$  and  $B$ . For example, given the strings ALGORITHM and ALTRUISTIC, your algorithm would return 5, the length of the longest common subsequence ALRIT.
- (b) Describe a simple *recursive* algorithm to compute, given two sequences  $A[1..m]$  and  $B[1..n]$ , the length of the *shortest common supersequence* of  $A$  and  $B$ . For example, given the strings ALGORITHM and ALTRUISTIC, your algorithm would return 14, the length of the shortest common supersequence ALGTORUISTHIMC.

*You do not need to analyze the running time of your algorithms for parts (a) and (b), but you still need to justify correctness. We are not looking for the most efficient algorithms, but for algorithms with simple and correct recursive structure.*