Problem Set 2

This problem set is due **at 10:00pm** on **Wednesday, September 19th, 2018.**

Please make note of the following instructions:

- This assignment, like later assignments, consists of *exercises* and *problems*. **Hand in solutions to the problems only.** However, we strongly advise that you work out the exercises for yourself, since they will help you learn the course material. You are responsible for the material they cover.
- We require that the solution to the problems is submitted as a PDF file, **typeset on LaTeX**, using the template available in the course materials. Each submitted solution should start with your name, the course number, the problem number, your recitation section, the date, and the names of any students with whom you collaborated.
- We will often ask you to "give an algorithm" to solve a problem. Your write-up should take the form of a short essay. Start by defining the problem you are solving and stating what your results are. Then provide: (a) a description of the algorithm in English and, if helpful, pseudo-code; (b) a proof (or proof sketch) for the correctness of the algorithm; and (c) an analysis of the running time. We will give full credit only for correct solutions that are described clearly and convincingly.

EXERCISES (NOT TO BE TURNED IN)

Divide and Conquer

- Do Exercise 9.3-7 in CLRS on page 223.
- Do Exercise 9.3-9 in CLRS on page 223.

Fast Fourier Transform

- Do Exercise 30.1-1 in CLRS on page 905.
- Do Exercise 30.1-4 in CLRS on page 905.
- Do Exercise 30.2-4 in CLRS on page 914.

Problem 2-1. Transportation in Line World [50 points]

Ben Bitdiddle is an architect in Line World. The government of Line World wants to build infrastructure to better connect the cities of the world in the face of increasing globalization. There exist n cities in Line World which all lie along the positive x-axis, and Ben has access to the location of each city in an unsorted list $x_1, ...x_n$.

(a) [20 points] The government tasks Ben with creating an airport in Line World such that the sum of the distances between the airport and each city is minimized. Design and analyze the time complexity of an algorithm to determine the location of such an airport. For full credit, this algorithm should run in $O(n)$ time.

(b) [20 points] Besides the locations of the cities, Ben now also has access to the population of each city $p_1, ...p_n$. The government wants Ben to build the airport now such that the airport is in a location that minimizes the total distance of the the airport from each civilian in the world. Design and analyze the time complexity of an algorithm to determine the location of such an airport. For full credit, this algorithm should run in $O(n)$ time. Partial credit can be received for algorithms in $O(n \log n)$ time.

(c) [10 points] Suffering from overpopulation, the Line World government decides to expand into a new dimension. Now, Ben has access to each city's location (x_i, y_i) and population p_i. However, due to construction, airplanes in line world can still only travel horizontally and vertically, parallel to the x and y axes respectively. Ben again wants to build the airport such that the airport is in a location that minimizes the total flight distance from each civilian in the world. Design and analyze the time complexity of an algorithm to determine the location (x_a, y_a) of such an airport. For full credit, this algorithm should run in the same time complexity as the previous question.

Problem 2-2. Binary strings [50 points] Alyssa P. Hacker has two binary strings (strings which contain only the characters '0' and '1'). Her first string A is of length $2n$ and her second string B is of length n. She wishes to find the locations of all substrings of A which have a Hamming distance from B which is less than or equal to d, for some nonnegative integer d. (The Hamming distance between two strings of the same length is defined as the number of positions at which they have different characters.) For example, when $A =$ '10101011', $B =$ '1010', and $d = 2$, the locations of all such substrings in A are indices 0, 2, and 4, since '1010' has a Hamming distance of 0 from B and '1011' has a Hamming distance of 1 from B.

(a) [5 points] Show an $O(n^2)$ solution for Alyssa's problem.

(b) [45 points] But can she do better? Show that there is an $O(n \log n)$ algorithm for the problem. (Hint: Find a good polynomial representation for strings. It may help to first think about the case when $d = 0$.)