

# CS 312: Algorithms

## Homework 7

Dan Sheldon

March 27, 2013

### Instructions

Complete all problems and submit by Tuesday, April 2. You may work together with other students, but *your written work must be your own*. I highly encourage you to attempt the problems first on your own, especially the simpler ones.

Please make sure to:

- Write your name on your submission
- Write the name of all students with whom you collaborated
- Cite any sources you used other than the textbook or course notes.

### Problems

1. (10 points) Given an array of  $n$  numbers, consider the problem of finding the contiguous subsequence of the input with the greatest possible sum. For example, in the array

$$\{31, -41, 59, 26, -53, 58, 97, -93, -23, 84\}$$

the maximum is achieved by summing the third through seventh elements, to get  $59 + 26 + (-53) + 58 + 97 = 187$ . When all numbers are positive, the entire array is the answer, while when all numbers are negative, the answer is the empty subsequence, which has sum zero.

- (a) (4 points) Give a simple, clear, and correct algorithm with running time  $\Theta(n^2)$  to find the maximum contiguous subsequence.
  - (b) (6 points) Now give a dynamic programming algorithm with running time  $\Theta(n)$  to solve the problem.
2. (10 points) Chapter 6 Exercise 1
  3. (10 points) You are competing with your friends to type text messages on your smartphone as quickly as possible. Here are the rules: you use two thumbs for texting and they start out on the bottom left and bottom right keys of the keyboard. To type a character, you move either thumb from its current key to the key you need to press, and it takes time proportional to the Euclidean distance between the keys. You can assume the following:
    - You can press any key with either thumb.
    - The keyboard has  $k$  keys arranged in a grid. (You may wish to think of the standard numeric keypad as an example: it has 12 keys for the digits 0–9 and the characters \* and #, and they are arranged in a  $3 \times 4$  grid.)

- Each character in your message takes exactly one key press.

Design an algorithm that takes an  $n$ -character message as input, and finds the fastest way to type it. In other words, your algorithm needs to decide which thumb to use to type each character, and it should minimize the total distance moved by your two thumbs. Try to use  $O(nk^3)$  time.

**References:** Problems 1 and 3 are adapted from *The Algorithm Design Manual* by Steve Skiena.