## Spring 2013 CS 401 Homework 3

State all necessary assumptions clearly. Due date: April 2, 2013 in class

- 1. The general form of a recurrence relation is: T(n) = aT(n/b) + f(n).
  - (a) Let f(n) = n. For each value of  $a \in \{2,3,4\}$  and for each value of  $b \in \{2,3,4\}$ , solve the recurrence relation. Express your answer in a 3-by-3 table.
  - (b) Repeat, assuming  $f(n) = \log n$ .
  - (c) Repeat, assuming  $f(n) = n \log n$ .
  - (d) Repeat, assuming  $f(n) = n^2$ .
- 2. Chapter 5, Problem 4
- 3. Chapter 5, Problem 5
- 4. Suppose you have a business which caters to Chicago and St. Louis. Each month, you can choose to either run your business from an office in Chicago, or from an office in St. Louis. In month *i*, you incur an operating cost of  $C_i$  if you run the business out of Chicago, and a cost of  $S_i$  if you instead run the business out of St. Louis. Each time you decide to switch between cities between two consecutive months, you incur a moving cost of M. Given a sequence of *n* months, a plan is a sequence of *n* locations (each one equal to either Chicago or St. Louis) such that the *i*<sup>th</sup> location indicates the city in which you will be based in the *i*<sup>th</sup> month. The cost of a plan is the sum of the operating costs for each of the *n* months, plus a moving cost *M* for each time you switch cities. The plan can begin in either city.

Your task is as follows: Given a value for M and sequences  $(C_1; C_2; ;;; C_n)$  and  $S_1; S_2; ;;; S_n)$ , give an efficient dynamic programming algorithm which returns the cost of an optimal plan for the n months in question.

5. In this problem, you will work on the task of word segmentation. Specifically, given a string of letters  $y = y_1y_2 \dots y_n$ , you would like to compute a segmentation of y. A segmentation of y is a partition of its letters into contiguous blocks of letters; each block corresponds to a word in the segmentation. For example, given y = meetateight, you might wish to split y into three blocks: "meet", "at", and "eight". To judge how good a block  $x = x_1x_2\dots x_k$  is, assume you are given a black box that, on input x, returns in constant time a real number quality(x) such that: A large positive value for quality(x) indicates x is close to an English word, and a large

The *total quality* of a segmentation is determined by adding up the qualities of each of its blocks.

negative number indicates x is far from an English word.

Your task is as follows: Give an efficient dynamic programming algorithm that takes a string y as input and computes a segmentation of maximum total quality (over all possible segmentations of y).

6. For the sequence alignment problem, modify the code on page 282 to also produce (output) the actual alignment that gives the optimal value of the alignment.

7. Let G = (V, E) be a directed graph with nodes  $v_1, v_2, \ldots v_n$ . We say that G is an ordered graph if it has the following properties. (i) Each edge goes from a node with a lower index to a node with a higher index. That is, every directed edge has the form  $(v_i, v_j)$  with i < j. (ii) Each node except  $v_n$  has at least one edge leaving it. That is, for every node  $v_i, i = 1, 2, \ldots, n-1$ , there is at least one edge of the form  $(v_i, v_j)$ . The length of a path is the number of edges in the path.

Given an ordered graph G, find the length of the longest path that begins at  $v_1$  and ends at  $v_n$ . Use a dynamic programming approach. What is the time complexity of your solution?