CS140: Algorithms

Fall 2001

Assignment 9

This assignment is due by the start of class on Wednesday 11/7/01. You may collaborate with your classmates to complete this assignment but you must reference your collaborators. In addition, you must do your own write-up. This assignment is worth 150 points (65 points if handwritten).

1. You win the candy store jackpot! Your prize is a bag full of the candy of your choice. The bag they let you fill is huge but, unfortunately, it is plastic and will break if you try to put in more than K pounds. Alas, you don't like candy. But you figure you can sell the candy at Platt and make big bucks. So your objective, in choosing the candy, is to maximize its value. The candy store has n different kinds of candy. Candy i costs p_i per pound and the store currently has x_i pounds of it. (Typically $x_i < K$.) Design an algorithm to determine an optimal selection of candy.

2. You are writing the HELP section of a software tool and decide to organize the help topics in a balanced binary search tree in order to minimize the search time. The help topics are {AVOCADO, BANANA, CARROT, DANDELION, ETCETERABERRY, FIZZLEWACKY, GRAPE, HUPPALA, ICKYPOO, JELLO, and KRUD}. Give a blanced binary search tree for this input.

   After many months of use you discover users rarely search for FIZZLEWACKY but often look for KRUD. And unfortunately, FIZZLEWACKY has a shorter search path than KRUD in your tree. Give a (not-necessarily-balanced) binary search tree for the help topics in which the search path for KRUD has length 1. (We'll define the length of the search path for a key as the number of queries needed to find that key.)

   You realize that your original tree minimizes the worst case search time but it might be better to minimize the expected search time. The expected search time is $\sum_i p_i \ell(w_i)$, where $w_1, \ldots w_n$ are the help topics, $\ell(w_i)$ is the length of the search path for w_i, and p_i is the probability that a user searches on the i-the topic.

   (a) Give a recursive definition for the minimum expected search time $E_{OPT}(w_1, \ldots, w_n, p_1, \ldots, p_n)$.

(b) Give a dynamic programming algorithm that computes E_{OPT}.

(c) Give a dynamic programming algorithm that produces a search tree with expected search time E_{OPT}.

(d) Analyze the running times of your algorithms.

NOTE: The tree you produce must be a binary search tree; i.e. the keys of nodes in the left (resp. right) subtree of the root are less than (resp. greater than) the key of the root.

3. M is an $n \times m$ matrix of distinct integers in which the rows are sorted in ascending order (smallest at left) as well as the columns (smallest at top). The numbering convention we'll follow is that $M(1,1)$ is the upper, left element of the matrix and $M(n,m)$ is the lower right element.

   (a) Construct a sample 3×4 matrix that is sorted as described.

   (b) Give the indices of the elements of M that are guaranteed to be strictly smaller than the element $M(k,\ell)$.

   (c) Give the indices of the elements of M that are guaranteed to be strictly larger than the element $M(k,\ell)$.

(d) Design an algorithm to search for an integer x in M. The algorithm can access M through queries of the form "How does $M(k, \ell)$ compare to x?" which is answered (a) "$M(k, \ell) = x$," (b) "$M(k, \ell) > x$," or (c) "$M(k, \ell) < x$." Your algorithm should return the location of x in M (the indices) or report that x is not in M.

To receive full credit your algorithm should make at most $m + n - 1$ queries in worst case. To receive half-credit give an $O(n + m)$ algorithm.

Hint: Find a first query such that a (b) or (c) response allows you to eliminate one row or one column from consideration.

4. Given a parenthesis-free expression over the integers and $\{+, -, *\}$ we want to parenthesize the expression so as to maximize its value. For example, the expression $-2 + 5 * 3 - 10$ can be parenthesized as $-2+((5*3)- 10)$, which evaluates to 3. It can also be parenthesized as $((-2 + 5) * 3) - 10$, which evaluates to -1.

(a) Parenthesize the following expressions so as to maximize their values.

i. $10 \ * \ -3 \ + 5$

ii. $3 \ - \ 4 \ - \ 2 \ * \ 5 \ + \ -1$

(b) Design an algorithm that takes as input an expression $v_1 o_1 v_2 o_2 \ldots o_{n-1} v_n$, where each v_i is an integer and each o_i is an operator from $\{+, -, *\}$, and parenthesizes it so as to maximize its values. Hint 1: Strengthen the inductive hypothesis. Hint 2: Dynamic programming.