CS 577: Introduction to Algorithms	10/11/2012

Homework 3

Instructors: Shuchi Chawla and Dieter van Melkebeek — Due: 10/23/2012

This assignment covers dynamic programming (chapter 6 in the textbook). See the first assignment for guidelines.

Ungraded problems

1. Problem 6.9 in the textbook (p. 320-321).

2. Martian currency has n different coins with integral denominations $a_1 < a_2 < \cdots < a_n$. For a given integer value A, your goal is to make change for a total amount of A using the fewest number of coins possible. Give an algorithm for this problem that runs in time polynomial in n and A.

3. You are given an arithmetic expression containing n integers and $n-1$ operators, each either $+$, $-$, or $\times$. Your goal is to perform the operations in an order that maximizes the value of the expression.

   For example:

   - For the expression $6 \times 3 + 2 \times 5$, the optimal ordering is to add the middle numbers first, then perform the multiplications: $((6 \times (3 + 2)) \times 5) = 150$.
   - For the expression $(-3) \times 3 + 3$, the optimal ordering is $(((-3) \times 3) + 3) = -6$.
   - For the expression $(-3) \times 3 - 3$, the optimal ordering is $((-3) \times (3 - 3)) = 0$.

   Give a polynomial-time algorithm to find the maximum possible value of the given expression.

Graded problems

4. (5 points) Problem 6.26 in the textbook (p. 333). For full credit the running time of your algorithm should be polynomial in n, that is, it should be independent of S.

5. (5 points) Problem 6.24 in the textbook (p. 331-332).

6. (5 points) Problem 6.28 in the textbook (p. 334). For full credit the running time of your algorithm should be polynomial in n, that is, it should be independent of D.

Extra-credit problems

7. Give an $O(n \log n)$ algorithm for problem 6.

8. There is a famous joke-riddle for children:

   > Three turtles are crawling along a road. One turtle says: "There are two turtles ahead of me." The other turtle says: "There are two turtles behind me." The third turtle says: "There are two turtles ahead of me and two turtles behind me." How could this have happened? The answer is – the third turtle is lying!

In this problem you have n turtles crawling along a road. Some of them are crawling in a group, so that they do not see members of their group neither ahead nor behind them. Each turtle makes a statement of the form: "There are a_i turtles crawling ahead of me and b_i turtles crawling behind me."

Your task is to find the minimal number of turtles that must be lying. More formally, let x_i denote the position along the road of turtle i, $1 \leq i \leq n$. Some turtles may be at the same position. Turtle i tells the truth if and only if a_i is the number of turtles j such that $x_j > x_i$ and b_i is the number of turtles j such that $x_j < x_i$. Otherwise, turtle i is lying.

Give an algorithm that solves the problem in $O(n \log n)$ time.