# CS 577: Introduction to Algorithms (Section 1) — Homework 4

**Out: 10/29/14** — **Due: 11/12/14**

## Ground Rules

See homework 1.

## Ungraded problems

1. Martian currency has $n$ different coins with integral denominations $a_1, \cdots, a_n$. 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 pseudo polynomial time in the size of the input and the output.

2. 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.

3. Problem 6.24 in the textbook (p. 331-2).

## 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)** In a tribute to Dr. Seuss, here is a question based on his story "Yertle the turtle". You don't need to know the story to solve the question.

   Mack, in an effort to avoid being cracked, has enlisted your advice as to the order in which turtles should be dispatched to form Yertle's throne. Each of the turtles ordered by Yertle has a different weight and strength. Your task is to build the largest stack of turtles possible such that for every turtle in the stack, its strength is no less than the total weight that it carries above it.

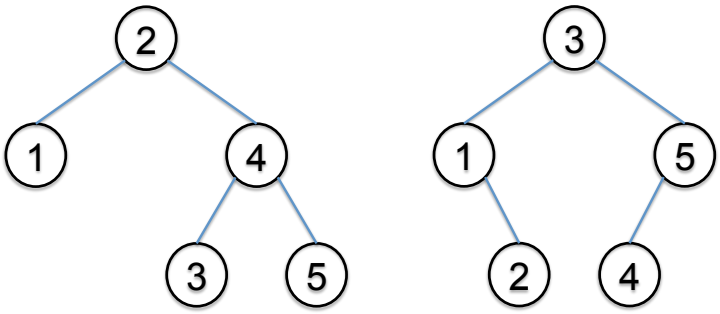
   You are given the weight and the strength of each turtle. The strength is the turtle's overall carrying capacity, including its own weight. That is, a turtle weighing 300 units with a strength of 1000 units could carry other turtles weighing a total of 700 units on its back.

   Your algorithm should output the maximum number of turtles that can be stacked without exceeding the strength of any one. It should run in time $O(n^2)$, where $n$ denotes the number of turtles given. You will be given partial credit for an algorithm that runs in time polynomial in $n$ and the weights or strengths of the turtles.

6. **(5 points)** One way to search efficiently is to use binary search trees. A binary search tree is a binary tree where each node is labeled with one of the elements and the labels satisfy the following property — the label of every node is larger than the label of every node in its left subtree, and smaller than the label of every node in its right subtree. Two examples of binary search trees over the elements $\{1, 2, 3, 4, 5\}$ are given below.

   Given the frequencies $f_1, \cdots, f_n$ of searching for elements $1, \cdots, n$, the average cost of search is equal to $\sum_i f_i \ell_i$ where $\ell_i$ is the "depth" of element $i$, or the length of the path from the root to the node labeled $i$.

   Give an algorithm for constructing the optimal binary search tree (minimizing the average cost of search) given $n$ elements $1, \cdots, n$ and their frequencies $f_1, \cdots, f_n$. Your algorithm should run in time polynomial in $n$.


2
1
4
3
5
3
1
5
2
4