Algorithms, Winter 2010-11, Homework 2
due Friday 12/17/10, 4pm

Problem 1

Consider the following recurrence:

$$\begin{aligned} T(n) &\leq 4T(\lfloor n/2 \rfloor) + 5n^2 \text{ for } n > 1 \\ T(1) &\leq 5 \end{aligned}$$

Find a constant $d > 0$ such that $T(n) \leq dn^2 \log n + dn^2$ for every $n \geq 1$. Use strong mathematical induction to show that for the chosen d the inequality $T(n) \leq dn^2 \log n + dn^2$ holds for every $n \geq 1$.

Problem 2

There are n stacks of DVD's forming a line, we number them from left to right: the leftmost stack is number 1, the rightmost stack is number n. A robot needs to rearrange the DVD's so that all stacks are of the same height (i.e., contain the same number of DVD's). The robot can perform only this move: go to the i-th stack where $i \in \{1, 2, \ldots, n\}$, pick any number of DVD's and move them to the stack immediately to the right (the $(i+1)$-st stack, if $i < n$) or immediately to the left (the $(i-1)$-st stack, if $i > 1$). Design an $O(n)$ algorithm that finds the minimum number of moves the robot needs to make in order to get all stacks to contain the same number of DVD's. Your algorithm should also list the moves the robot needs to make. If it is not possible to achieve equal heights, your algorithm should say so. Argue both the correctness and the time complexity of your algorithm.

For example, if we start with stacks of 2, 6, 3, and 1 DVD's, the robot needs to move 1 DVD from the 2nd to the 1st stack, 2 DVD's from the 2nd to the 3rd stack, and 2 DVD's from the 3rd to the 4th stack.

Problem 3

(a) We are given an array of integers $A[1..n]$. We would like to determine whether there exists an integer x that occurs in A more than $n/2$ times (i. e., whether A has a majority element). Design an algorithm that runs in time $O(n)$ and argue its correctness and running time estimate.

Example: For $A = [3, 1, 2]$ the answer is NO. For $A = [3, 1, 3]$ the answer is YES.

(b) We are given an array of integers $A[1..n]$. We would like to determine whether there exists an integer x that occurs in A more than $n/3$ times. Design an algorithm that runs in time $O(n)$ and argue its correctness and running time estimate.