CSE/MAT 373 Spring 2009 Analysis of Algorithms

February 8, 2009

- Homework is due **February 19th** STRICTLY before class.
- You write down the solution clearly, and if possible, succinctly. Avoid too many details – a succinct and clean proof is the best.
- Read the "Course Policies" on the class web-page, and strictly follow the guidelines therein.

Homework 1

1. (30 points) Exercise 0.1 of [DPV]. Here and below, [DPV] refers to the online version of our course textbook.
2. (15 points) Exercise 0.2 of [DPV].
3. (15 points) Exercise 0.3 of [DPV].
4. (10 points) Prove for all $k \geq 1$ and all sets of real-number constants $\{a_0, a_1, a_2, \ldots, a_{k-1}, a_k\}$:

$$a_k n^k + a_{k-1} n^{k-1} + \ldots + a_1 n + a_0 = \mathcal{O}(n^k).$$

5. (20 points) Suppose that we are given a stack of n elements which we would like to sort, by returning a stack containing the records in sorted order (with the smallest value on top). We are allowed to use only the following operations to manipulate the data:

   - `Pop-push(`s_1, s_2`)` — pop the top item from stack s_1 and push it onto stack s_2.
   - `Compare(`s_1, s_2`)` — test whether the top element of stack s_1 is less than the top element of stack s_2.

   Give a $\Theta(n^2)$ sorting algorithm using just these operations and **two** stacks, where you are allowed to temporarily pack elements in a constant number of additional registers (hint: use insertion sort).

6. (10 points) Let $A[1..n]$ be an array of real numbers. Design an algorithm to perform any sequence of the following operations:

   - *Add(i, y)* — add the value y to the i^{th} number.
   - *Partial-sum(i)* — return the sum of the first i numbers, i.e., $\sum_{j=1}^{i} A[j]$.

   There are no insertions or deletions, the only change is to the values of its numbers. Each operation should take $\mathcal{O}(\log n)$ steps. You may use one additional array of size n as a work space.