University of Washington — January 12, 2000
Department of Computer Science and Engineering
CSE 521, Winter 2000

Homework 2, **Due Wednesday, January 19, 2000**

Reading Read chapter 16 for dynamic programming.

Problem 1:
Let

$$T(n) = \sum_{i=1}^{k} a_i T\left(\left\lfloor \frac{n}{b_i} \right\rfloor\right) + cn$$

with

$$\sum_{i=1}^{k} \frac{a_i}{b_i} = x < 1.$$

Prove that $T(n)$ is $O(n)$.

Problem 2:
Consider the following algorithm for sorting. The call *SlowSort*(A, i, j) sorts the items $A[i] \ldots A[j]$.

SlowSort(A, i, j)
  if $i \geq j$ **then return**
  $m := \lfloor (i + j)/2 \rfloor$;
  SlowSort(A, i, m);
  SlowSort$(A, m + 1, j)$;
  if $A[m] > A[j]$ **then**
    exchange $A[j]$, $A[m]$;
  SlowSort$(A, i, j - 1)$;

a) Give an informal proof that the algorithm sorts correctly.

b) Give a recurrence relation for the run time $T(n)$ to sort n items.

c) Show that $T(n)$ is $O(n^{(\log n)/2})$ and $T(n)$ is $\Omega(n^{(\log n)/4})$.

(This algorithm is due to Broder and Stolfi [SIGACT News, Fall, 1984]).

Problem 3:
CLR, Page 192, Exercise 10.3-8.

Problem 4:
CLR, Page 193, Problem 10-2.

Problem 5:
CLR, Page 744, Exercise 31.2-2.

Problem 6:
CLR, Page 908, Exercise 35.3-6.

Problem 7:
(Challenge) CLR, Page 914, problem 35-3. Hint - start with a point on the convex hull.