Problem Set 1

This problem set is due in lecture on **Monday, September 18, 2006**. The homework lab for this problem set will be held 2–4 P.M. on Sunday, September 17, 2006, in room 32-124.

**Reading:** Chapters 1–4 excluding Section 4.4.

Both exercises and problems should be solved, but *only the problems* should be turned in. Exercises are intended to help you master the course material. Even though you should not turn in the exercise solutions, you are responsible for material covered in the exercises.

Mark the top of each sheet with your name, the course number, the problem number, your recitation section, the date and the names of any students with whom you collaborated.

You will often be called upon to “give an algorithm” to solve a certain problem. Your write-up should take the form of a short essay. A topic paragraph should summarize the problem you are solving and what your results are. The body of the essay should provide the following:

1. A description of the algorithm in English and, if helpful, pseudo-code.
2. At least one worked example or diagram to show more precisely how your algorithm works.
3. A proof (or indication) of the correctness of the algorithm.
4. An analysis of the running time of the algorithm.

Remember, your goal is to communicate. Full credit will be given only to correct solutions which are described clearly. Convoluted and obtuse descriptions will receive low marks.

**Exercise 1-1.** Do Exercise 2.3-6 on page 37 in CLRS.

**Exercise 1-2.** Do Exercise 3.1-4 on page 50 in CLRS.

**Exercise 1-3.** Do Exercise 3.2-4 on page 57 in CLRS.

**Exercise 1-4.** Do Problem 4.3-4 on page 75 of CLRS.

**Problem 1-1. Asymptotic Notation**

For each of the following relationships, find two nonnegative functions $f(n)$ and $g(n)$ that satisfy it. If no such functions exist, write “NONE” and briefly justify your answer.
(a) \( f(n) = O(g(n)) \) and \( g(n) = O(f(n)) \).

(b) \( f(n) = o(g(n)) \) and \( g(n) = o(f(n)) \).

(c) \( f(n) = \omega(f(n)) \)

(d) \( f(n) = \Omega(g(n)) \) and \( g(n) = \omega(f(n) + g(n)) \).

(e) \( f(n) = \Theta(1) \) and \( g(n) = o(f(n)) \)

(f) \( f(n) = O(1) \) and \( g(n) = o(f(n) + g(n)) \).

Problem 1-2. Recurrences

Give asymptotic upper and lower bounds for \( T(n) \) in each of the following recurrences. Assume that \( T(n) \) is a nonnegative constant for \( n \leq 10 \). Make your bounds as tight as possible, and justify your answers.

(a) \( T(n) = 3T(n/3) + n \)

(b) \( T(n) = 11T(n/7) + n^2 \)

(c) \( T(n) = 16\left\lceil n/2 \right\rceil + \left(\begin{array}{c} n \\ 3 \end{array}\right) \lg^4 n \)

(d) \( T(n) = 4/3 T\left(\begin{array}{c} 3 \\ 4 \end{array} n \right) + 4/3 n \)

(e) \( T(n) = 2T(n/2) + \lg(n!) \)

(f) \( T(n) = T(n - 10) + n \)

(g) \( T(n) = 2T(n/4) + 6.046 \sqrt{n} \)

(h) \( T(n) = T(n/3) + T(n/4) + n \)

(i) \( T(n) = 3T(n^{1/3}) + \lg \lg n \)

Problem 1-3. Lightest Segment Problem

Given an input array \( A[1 \ldots n] \) of numbers, a segment of \( A \) is a consecutive interval of one or more elements from \( A \) of the form \( A[i \ldots j] \) where indices \( i \) and \( j \) satisfy \( 1 \leq i \leq j \leq n \). The weight of the segment \( A[i \ldots j] \) is defined to be \( \sum_{k=i}^{j} A[k] \). The absolute weight of \( A[i \ldots j] \) is \( \sum_{k=i}^{j} |A[k]| \).

The absolute weight is always nonnegative. In the lightest segment problem, the goal is to find the minimum absolute weight of a segment in the given array \( A[1 \ldots n] \).

(a) Give a simple \( O(n^3) \)-time algorithm for the lightest segment problem.

In the rest of this problem, we develop a more efficient solution using the divide-and-conquer paradigm.

(b) Give an \( O(\lg n) \)-time algorithm that, given a sorted array \( B \) of \( n \) numbers and given an integer \( x \), determines the minimum value attained by \(|B[i] + x|\) where \( i \) varies between 1 and \( n \).
(c) Give an $O(n \log n)$-time algorithm that, given two arrays $B_1[1..n]$ and $B_2[1..n]$ of numbers, determines the minimum value attained by $|B_1[i] + B_2[j]|$, where $i$ and $j$ vary between 1 and $n$. Hint: Use the algorithm from part (b).

We distinguish three different kinds of segments $A[i..j]$. Define $m$ to be $\lfloor n/2 \rfloor$. If $i, j \leq m$, we call the segment left. If $i, j > m$, we call the segment right. Otherwise, $i \leq m$ and $j > m$, and we call the segment middle.

(d) Give an $O(n \log n)$-time algorithm that, given an array $A[1..n]$ of numbers, determines the minimum absolute weight of a middle segment $A[i..j]$ (where $1 \leq i \leq m$ and $m < j \leq n$).

Hint: The weight of a middle segment $A[i..j]$ can be written as the sum of the weights of two subsegments, $A[i..m]$ and $A[m+1..j]$. There are only $O(n)$ such subsegments; avoid recomputing their weights.

(e) Give an $O(n \log^2 n)$-time divide-and-conquer algorithm for the lightest segment problem.

(f) **Bonus Part.** This part is optional. If you solve this part, you do not need to write up your solutions to the other parts of this problem.

Give an $O(n \log n)$-time algorithm for the lightest segment problem.