Exercises I

Note that to *give an algorithm* means not only to describe the algorithm, but also to analyze its running time.

I-1 Let $T(n) = M$ for $n \leq M$ for some constant $M \geq 1$ independent of n, and $2\,T(n) = 2\,T(n/2) + 3\,T(n/3) + n$ otherwise. Show that $T(n) = \Theta(n \log n)$.

I-2 (**CLRS 2.3-7**) Describe a $\Theta(n \log n)$-time algorithm that, given a set S of n integers and another integer x, determines whether or not there exist two elements in S whose sum is exactly x.

I-3 (**CLRS 2-4 Inversions**) Let $A[1..n]$ be an array of n distinct numbers. If $i < j$ and $A[i] > A[j]$, then the pair (i, j) is called an *inversion* of A.

- **a.** List the five inversions of the array $(2, 3, 8, 6, 1)$.
- **b.** What array with elements from the set $\{1, 2, \dots, n\}$ has the most inversions? How many does it have?
- **c.** What is the relationship between the running time of insertion sort and the number of inversions in the input array? Justify your answer.
- **d.** Give an algorithm that determines the number of inversions in any permutation of n elements in $\Theta(n \log n)$ worst-case time. (*Hint:* Modify merge sort.)

I-4 (**CLRS 4.2-6**) How quickly can you multiply a $kn \times n$ matrix by an $n \times kn$ matrix, using Strassen's algorithms as a subroutine? Answer the same question with the order of input matrices reversed.

I-5 (**CLRS 4.2-7**) Show how to multiply the complex numbers $a + bi$ and $c + di$ using only three multiplications of real numbers. The algorithm should take a, b, c, and d as input and produce the real component $ac - bd$ and the imaginary component $ad + bc$ separately.