1.1-1

Using Figure 1.2 as a model, illustrate the operation of INSERTION-SORT on the array $A = \langle 31, 41, 59, 26, 41, 58 \rangle$.

Do 1.1-1 for Bubble, Insertion and Merge sorts but with inversed array.

1.2-3

Consider the problem of determining whether an arbitrary sequence $\langle x_1, x_2, \ldots, x_n \rangle$ of n numbers contains repeated occurrences of some number. Show that this can be done in $\Theta(n \lg n)$ time, where $\lg n$ stands for $\log_2 n$.

1.2-4

Consider the problem of evaluating a polynomial at a point. Given n coefficients $a_0, a_1, \ldots, a_{n-1}$ and a real number x, we wish to compute $\sum_{i=0}^{n-1} a_i x^i$. Describe a straightforward $\Theta(n^2)$-time algorithm for this problem. Describe a $\Theta(n)$-time algorithm that uses the following method (called Horner's rule) for rewriting the polynomial:

$$\sum_{i=0}^{n-1} a_i x^i = (\cdots(a_{n-1}x + a_{n-2})x + \cdots + a_1)x + a_0 .$$

1.2-5

Express the function $n^3/1000 - 100n^2 - 100n + 3$ in terms of Θ-notation.

1.3-3

Use mathematical induction to show that the solution of the recurrence

$$T(n) = \begin{cases} 2 & \text{if } n = 2 , \\ 2T(n/2) + n & \text{if } n = 2^k, k > 1 \end{cases}$$

is $T(n) = n \lg n$.

1.3-4

Insertion sort can be expressed as a recursive procedure as follows. In order to sort $A[1\,..\,n]$, we recursively sort $A[1\,..\,n-1]$ and then insert $A[n]$ into the sorted array $A[1\,..\,n-1]$. Write a recurrence for the running time of this recursive version of insertion sort.

2-1 Asymptotic behavior of polynomials

Let

$$p(n) = \sum_{i=0}^{d} a_i n^i ,$$

where $a_d > 0$, be a degree-d polynomial in n, and let k be a constant. Use the definitions of the asymptotic notations to prove the following properties.

a. If $k \geq d$, then $p(n) = O(n^k)$.

b. If $k \leq d$, then $p(n) = \Omega(n^k)$.

c. If $k = d$, then $p(n) = \Theta(n^k)$.

d. If $k > d$, then $p(n) = o(n^k)$.

2.1-1

Let $f(n)$ and $g(n)$ be asymptotically nonnegative functions. Using the basic definition of Θ-notation, prove that $\max(f(n), g(n)) = \Theta(f(n) + g(n))$.

3.2-1

Show that $\sum_{k=1}^{n} 1/k^2$ is bounded above by a constant.

3.2-2

Find an asymptotic upper bound on the summation

$$\sum_{k=0}^{\lfloor \lg n \rfloor} \lceil n/2^k \rceil .$$

3.2-3

Show that the nth harmonic number is $\Omega(\lg n)$ by splitting the summation.

3.2-4

Approximate $\sum_{k=1}^{n} k^3$ with an integral.

4.1-1

Show that the solution of $T(n) = T(\lceil n/2 \rceil) + 1$ is $O(\lg n)$.

4.2-1

Determine a good asymptotic upper bound on the recurrence $T(n) = 3T(\lfloor n/2 \rfloor) + n$ by iteration.

4.2-3
Draw the recursion tree for $T(n) = 4T(\lfloor n/2 \rfloor) + n$, and provide tight asymptotic bounds on its solution.

7.1-1
What are the minimum and maximum numbers of elements in a heap of height h?

7.2-5
Show that the worst-case running time of HEAPIFY on a heap of size n is $\Omega(\lg n)$. (*Hint:* For a heap with n nodes, give node values that cause HEAPIFY to be called recursively at every node on a path from the root down to a leaf.)

8.1-1
Using Figure 8.1 as a model, illustrate the operation of PARTITION on the array $A = \langle 13, 19, 9, 5, 12, 8, 7, 4, 11, 2, 6, 21 \rangle$.

Do 8.1-1 for both for Heap and Quick sorts, you can skip some similar steps if you feel comfortable with procedure.

FINAL PROBLEM - Implement Merge, Heap and Quick sorts.