Problems

1. Rewrite the insertion sort algorithm to sort into nonincreasing instead of nondecreasing order.

2. Consider the problem of adding two n-bit binary integers, stored in two n-element arrays A and B. The sum of the two integers should be stored in binary form in an (n+1)-element array C. Write pseudocode for this problem.

3. Illustrate the operation of merge sort on the array A=(3, 41, 52, 26, 38, 57, 9, 49).

4. 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.

5. Suppose you have algorithms with the six running times listed below. (Assume these are the exact number of operations performed as a function of the input size n.) Suppose you have a computer that can perform $10^{10}$ operations per second, and you need to compute a result in at most an hour of computation. For each of the algorithms, what is the largest input size n for which you would be able to get the result within an hour?

(a) $n^2$  (b) $n^3$  (c) $100n^2$  (d) $n \log n$  (e) $2^n$  (f) $n!$

6. Take the following list of functions and arrange them in ascending order of growth rate. That is, if function $g(n)$ immediately follows function $f(n)$ in your list, then it should be the case that $f(n)$ is $O(g(n))$.

$f_1(n) = n^{2.5}$  $f_2(n) = (2n)^{1/2}$  $f_3(n) = n+10$  $f_4(n) = 10^n$  $f_5(n) = 100^n$  $f_6(n) = n^2 \log n$

7. Using the master method, find solutions of the following recurrences.

(a) $T(n) = 4T(n/3) + n$  (b) $T(n) = 4T(n/2) + n^2$  (c) $T(n) = T(n-1) + n$

8. Illustrate the operation of quick sort on the array A=(3, 41, 52, 26, 38, 57, 9, 49).

9. Given two integers a and b, write an iterative algorithm to find greatest common divisors of these two numbers.

10. Given two integers a and b, write a recursive algorithm to find greatest common divisors of these two numbers.