Cairo University
Faculty of Engineering
Computer Engineering Department

Algorithms
Sheet #1

1. Suppose we are comparing implementations of insertion sort and merge sort on the same machine. For inputs of size n, insertion sort runs in $8n^2$ steps, while merge sort runs in 64n lg n steps. For which values of n does insertion sort beat merge sort?

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 . State the problem formally (Input & Output) and write pseudocode for adding the two integers.

3. Describe a (n lg 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.

4. Give the time complexity order for the following:
   a) $T(n) = T(n-1) + n$
   b) $T(n) = T(n/2) + 1$
   c) $T(n) = 4T(n/3) + n$
   d) $T(n) = 2T(n/4) + n^2$

5. What is the running time of HEAPSORT on an array A of length n that is already sorted in increasing order? What about decreasing order?

6. Show that the running time of QUICKSORT is $\Theta(n^2)$ when the array A contains distinct elements and is sorted in decreasing order.

7. Show how to sort n integers in the range 0 to $(n^3 - 1)$ in O(n) time.

8. Demonstrate what happens when we insert the keys 5; 28; 19; 15; 20; 33; 12; 17; 10 into a hash table with collisions resolved by chaining. Let the table have 9 slots, and let the hash function be h(k) = k mod 9.

9. Consider inserting the keys 10;22;31;4;15;28;17;88;59 into a hash table of length m = 11 using open addressing with the auxiliary hash function $h'(k) = k$. Illustrate the result of inserting these keys using linear probing, using quadratic probing with $c_1 = 1$ and $c_2 = 3$, and using double hashing with $h_1(k) = k$ and $h_2(k) = 1 + (k \bmod (m - 1))$.