CSCI 303 Homework 1

Problem 1 (2.1-1):
Using Figure 2.2 as a model, illustrate the operation of INSERTION-SORT on the array $A = \langle 31, 41, 59, 26, 41, 58 \rangle$.

Problem 2 (2.2-1):
Express the function $n^3/1000 - 100n^2 - 100n + 3$ in terms of Θ-notation.

Problem 3 (Derived from 2.2-2):
Consider sorting n numbers stored in array A by first finding the smallest element of A and exchanging it with the element in $A[1]$. Then find the second smallest element of A, and exchange it with $A[2]$. Continue in this manner for the first $n-1$ elements of A. Write pseudocode for this algorithm, which is known as SELECTION-SORT. Give the worst-case running times of selection sort in Θ-notation.

Problem 4 (2.3.1):
Using Figure 2.4 as a model, illustrate the operation of merge sort on the array $A = \langle 3, 41, 52, 26, 38, 57, 9, 49 \rangle$.

Problem 5 (Derived from 1.2-2):
Suppose we are comparing two sorting algorithms. Suppose that for all inputs of size n, the first algorithm runs in $8n^2$ seconds, while the second algorithm runs in $64n \lg n$ seconds. For which values of n does the first algorithm beat the second algorithm?

Problem 6 (1.2-3):
What is the smallest value of n such that an algorithm whose running time is $100n^2$ runs faster than an algorithm whose running time is 2^n?

Problem 7 (1.2-3):
Assume that a new Intel processor can execute 10^{15} operations per second. You have two algorithms that test whether a number is prime or not. The first algorithm uses $100n^2$ operations for a number with n decimal digits. The second uses 2^n operations for a number with n decimal digits. Using the first algorithm, how many seconds would the Intel processor take to determine whether a 1000 decimal digit number is prime? Using the second algorithm, how many seconds would the Intel processor take to determine whether a 1000 decimal digit number is prime?

Problem 8 (1-1 Comparison of running times):
For each function $f(n)$ and time t in the following table, determine the largest size n of a problem that can be solved in time t, assuming that the algorithm to solve the problem takes $f(n)$ microseconds.

	1 second	1 minute	1 hour	1 day	1 month	1 year	1 century
$\lg n$							
$\sqrt{n}$							
n							
$n \lg n$							
n^2							
n^3							
2^n							
$n!$							

Problem 9 (Derived from 3.1-2):
Show that for all numbers a and b, $(n+a)^b = \Theta(n^b)$.

Problem 10 (3.1-4):
Is $2^{n+1} = O(2^n)$? Is $2^{2n} = O(2^n)$?

Problem 11 (Not in book):
In the table below, write O if $f(n) = O(g(n))$, Ω if $f(n) = \Omega(g(n))$, and Θ if $f(n) = \Theta(g(n))$.

$g(n) \downarrow \quad f(n) \rightarrow$	$5n+3$	3.14×10^8	$8n^2 \lg^4 n + n^3$	2^n	$5n^3 + 4n^2 \lg n + 3n$	$12n + \lg^8 n$
$5n+3$						
3.14×10^8						
$8n^2 \lg^4 n + n^3$						
2^n						
$5n^3 + 4n^2 \lg n + 3n$						
$12n + \lg^8 n$						