Homework #1: Due 8/17/17 by 11:55pm*

1. a) Using Figure 2.4 in CLRS as a model, illustrate the operation of merge sort on the array $A = \langle 3, 51, 15, 9, 36, 56, 49, 2\rangle$.
   Solution. Your solution here.

   b) Prove that $o(g(n)) \cap \omega(g(n))$ is the empty set.
   Solution. Your solution here.

2. Fill the following blanks with the notations O, o, Ω, ω, or Θ. Use o if both O and o are applicable. Use ω if both Ω and ω are applicable. Use Θ if both Ω and O are applicable. To distinguish O and o, write "big-O" for O.

   a) $n =$ ________ $(2n)$.

   b) $0.2n =$ ________ $(2 \log n)$.

   c) $2^n =$ ________ (n^6).

   d) $2n \log n =$ ________ (n^2).

   Solution. Your solution here.

3. Give the running time at each step of the following high-level description of an algorithm, along with the total running time, briefly justifying each step. The input is an array with n items. The array is *scanned left-to-right* (that is, we have a for loop from $i = 0$ to $n - 1$), and at each iteration we do the following:

   a) Set $A[j] = A[j] \mod A[i]$ for all $0 \leq j \leq n - 1$.

   b) Sort the resulting array A.

   Solution. Your solution here.

4. One student of 122A proposes three variants of MergeSort. He/She describes the algorithms as follows. Suppose the input array is of size n.

*Last update August 11, 2017

a) MergeSortA: it divides the problem into 5 subproblems of half the size, solves each subproblem recursively and then combines the results in linear time.

b) MergeSortB: it divides the problem into 2 subproblems of size $n-1$, solves each subproblem recursively and then combines the results in constant time.

c) MergeSortC: it decides the problem into 9 subproblems of size $n/3$, solves each subproblem recursively and then combines the results in $O(n^2)$ time.

Give detailed steps to obtain the asymptotic running time (in big-O notation) of each algorithm. Which algorithm is the best? And explain your reason.

Solution. Your solution here.

5. The Fibonacci numbers $F_0, F_1, F_2, \ldots,$ are defined by the following rule:

$$F_0 = 0, F_1 = 1, F_n = F_{n-1} + F_{n-2}.$$

In this problem, we will see that this sequence grows exponentially fast. We will also establish some bounds on its growth.

a) Use induction to prove that $F_n \geq 2^{0.5n}$ for $n \geq 6$.

b) Find a constant $c < 1$ such that $F_n \leq 2^{cn}$ for all $n \geq 0$. Show that your answer is correct.

Solution. Your solution here.