**Assignment # 2**

**Please submit a report on your findings.**

The class AnalysisAssignment contains three methods called alg1, alg2 and alg3. The idea is to both theoretically and experimentally get a hint of the behaviors of the algorithms. For this exercise, perform the following:

1. Determine the growth function of the **alg1** method and the complexity of the algorithm.
2. What do **alg1** and **alg2** performs on the array passed as parameter (aka the “hidden task”)**?** To accomplish this, create a main method, declare an array of int numbers, fill it using the fillArray method, print the contents of the array before and after the execution of each of them (you may want to copy the contents of the first array to a second array before invoking **alg1** or **alg2**. Use copyArray to accomplish this).
3. **alg1** and **alg2** both returns integers that are the number of operations performed on each of these methods when they are invoked. Based on these numbers, answer the following questions:
4. For **alg1** how different is this number from the growth function that you obtained on (a)? To accomplish this try arrays of different sizes, say from 10 to 1000 on increments of 10, create a table with columns n , T(n) and the and the value that is returned by **alg1 .** Write your conclusions.
5. For **alg2** what is an approximation of its computational complexity? To accomplish this, try running alg2 with arrays of different sizes, say from 100 to 20000 in increments of 100, and given the Big-oh complexities that we saw in class (log(n), n, n\*log(n), n^2, n^3 , …) which of them better approximates running time the behavior of **alg2**? How did you reach this conclusion?
6. If you had to choose between alg1 and alg2 to perform the “hidden” task, which one would you use? To accomplish this task, run alg1 with an array, alg2 with a copy of the array that you pass to alg1, and print the number of operations for **alg1** and **alg2** for different sizes of the array.