**Introduction to Algorithms – Sort and Search**

**CSC212 Intermediate Object Oriented Programming**

**Professor: Dr. Julian Achim**

**By Odiscious Dozier**

**Trident University**

**Case 04**

In this Case Assignment, you are required to implement the Selection Sorting algorithm. You Java program will ask the user to input 10 numbers and your program stores these ten numbers into array and sort these 10 numbers in descending order (largest number to smallest number). Your program will output the numbers in the sorted array.

In this assignment I chose to use the generic array list working with the Integer class; i.e., ArrayList<Integer>. The interesting thing I’ve noticed here is that this object has accessors and mutator(s) (get and set) to access or modify its indexes and values whereas an array would simply use square brackets. The iteration cost is the same as well as the allocation of memory; so, there’s no runtime difference.

The selection sorting algorithm has a runtime cost of O(n^2) and is not a good choice for sorting arrays of large ‘n’.

An observation made during this particular exercise is that the inner for loop can start at either the outer for loop’s value or that value plus one; for example, let i = 0 for the outer loop and let (j = i) or (j = i + 1) for the inner loop. This exists in my code because, the conditional ‘if’ statement allows for it.

Additionally, in exploring the nature of the insertion sort and the selection sort it’s interesting to see how the same array is partitioned between sorted and unsorted sections. This portioning is very efficient with arrays of large ‘n’ and keeps costs, time as a function T(n) and memory allocation, at a minimum.