

**CSC 323 Algorithm Design and Analysis**  
**Fall 2019**  
**Instructor: Dr. Natarajan Meghanathan**

**Project 3: Recursive Algorithm to Find the Minimum Integer in an Array and its Space-Time Tradeoff Performance Comparison Analysis with an Iterative Algorithm**

**Due by: Oct. 8th, 11.59 PM (in Canvas)**

Design and implement a recursive algorithm to find the minimum integer in an array of integers. The recursive algorithm should be in-place (i.e., its memory requirements should not grow with the input size). You will also implement an iterative version of the algorithm to find the minimum integer in an array of integers and compare the performance of the recursive vs. iterative versions.

Run the implementations for 100 times (automated via a loop) for each of the following values of array size and average the results of the run-time (in milliseconds or nanoseconds, if the former is a very small value) for the recursive and iterative versions.

Array Size: 10,  $10^2$ ,  $10^3$ ,  $10^4$ ,  $10^5$

The value of the elements in the array can range from 0 to 9999.

**Submission:**

(1 - 45 pts) Submit a separate file featuring the code for the implementations of the recursive (30 pts) and iterative algorithms (15 pts)

(2 - 55 pts) Submit a single PDF file containing the following:

- (a - 15 pts) Pseudo code for the recursive algorithm and the explanation of its working
- (b - 8 pts) Justify the space-complexity of the algorithm is  $\Theta(1)$ , i.e., it is in-place.
- (c - 12 pts) Analyze the time-complexity of the algorithm and show that it is  $\Theta(n)$  for an array of size 'n'
- (d - 10 pts) Excel chart comparing the average run-time of the recursive vs. iterative algorithms for the different array sizes (consider plotting the X and Y values on a logarithmic scale, to the base 10)
- (e - 10 pts) Your interpretation of the results for the average run-time of the iterative and recursive versions.