CSC 323-01 Algorithm Design and Analysis, Fall 2019
Instructor: Dr. Natarajan Meghanathan

Quiz 4: Binary Search vs. Brute Force Search Algorithms for Finding a Local Minimum in a Two-Dimensional Array

Due by: Oct. 17th, 11.59 PM (in Canvas)

In this quiz, you will implement the binary search-based $\Theta(n)$ algorithm to determine the local minimum in an 'n' x 'n' two-dimensional array (as discussed in Module 2) and compare its run-time performance with that of a brute force $O(n^2)$ algorithm that searches for the local minimum element by element until one is found.

Note that both the binary search and the brute force search algorithms should stop once a local minimum is found.

You should create random two-dimensional arrays (with numRows = numCols) with unique elements in the range [1... numRows * numCols] for the following values of numRows (numCols) and determine the average execution time of the binary search and the brute force search algorithms by running 200 trials for each of the numRows (numCols) values. Determine the running times in nano seconds or milli seconds, as appropriate.

numRows (numCols) values: 4, 6, 8, 10, 15, 20, 25, 30, 35, 40, 50, 75, 100, 150, 200, 250, 300, 350

Plot the results with numRows in X-axis and the average execution times (in nanoseconds or milliseconds) of the binary search and the brute force search algorithms in the Y-axis.

Submission:
(1 - 60 pts) Code for the binary search and the brute force search algorithms submitted as separate files or single file, depending on your implementation.
(2 - 40 pts) A single PDF file featuring the following
- (15 pts): Your strategy/algorithm to generate a random two-dimensional array of unique elements; its time complexity and space-complexity analysis
- (10 pts): Excel plots of the numRows vs. average execution times of the two algorithms
- (15 pts): Your interpretation of the results.