# Assignment 2 – Complexity and Sorting Pt. 1

Due: Monday, February 5, 11:59PM

|  |  |  |
| --- | --- | --- |
| 1. | Write and test functions for finding the minimum, maximum, and mean (average) for unsorted and for sorted vectors. As a comment, what are the complexities of each algorithm in Big-O notation (assuming that sorting is not counted in the complexity of the algorithm)? | 30% |
|  |  |  |
| 2. | Write and test an algorithm to find the mode of a vector (the most frequently occurring element). If two or more elements occur most frequently, the function may return any of them as the mode. What is the complexity of your algorithm in Big-O notation?  For extra credit, make the function return a vector of all the modes if multiple numbers occur most often. For example, in the data 1, 2, 2, 3, 4, 4, 5, there are 2 modes – 2 and 4. | 40%    10% Extra |
|  |  |  |
| 2. | The implementation of Bubble Sort presented in class is a very simple one; it can be improved in several ways. The biggest improvement is from realizing that each time the outer loop runs, the largest number ends up in the right place. From that assumption, we can modify the algorithm to eliminate the need to check as many values. Another improvement can be made if we realize that if the inner loop completes without ever swapping two elements, then the vector is already sorted, and we can stop early.   Implement these 2 changes and compare the performance of your new function with the original Bubble Sort for vectors of size n = 10, 100, and 1000. Use the provided Timer class or something similar. | 30% |

Submit all project files to the assignment on MUOnline in a compressed (.zip) format. The submission should follow the naming convention:

CIT238\_*LastName\_FirstInitial\_*A2.zip