Data Structures and Algorithms - Assignment 1

Michael Thacker

Question 1.

The time complexity of my solution is O(*m*), where *m* is the number of integers between *start* and *end* inclusively. This is because the function must check every element of the row between these two columns in turn to check no element is larger than the previously found greatest one.

Question 3.

The upper bound for the time complexity of my matrixMaxValue function is O(nm). This is in the worst-case scenario when the maximum value in each row occurs at index 0. This means that no elements from the next row can be eliminated and every element (m elements) of each row (n rows) must be checked. Resulting in an O(n x m) time complexity, or O(nm).

My solution does decrease the average time complexity as, in the average case, the maximum value could be in any column allowing some values to be eliminated.