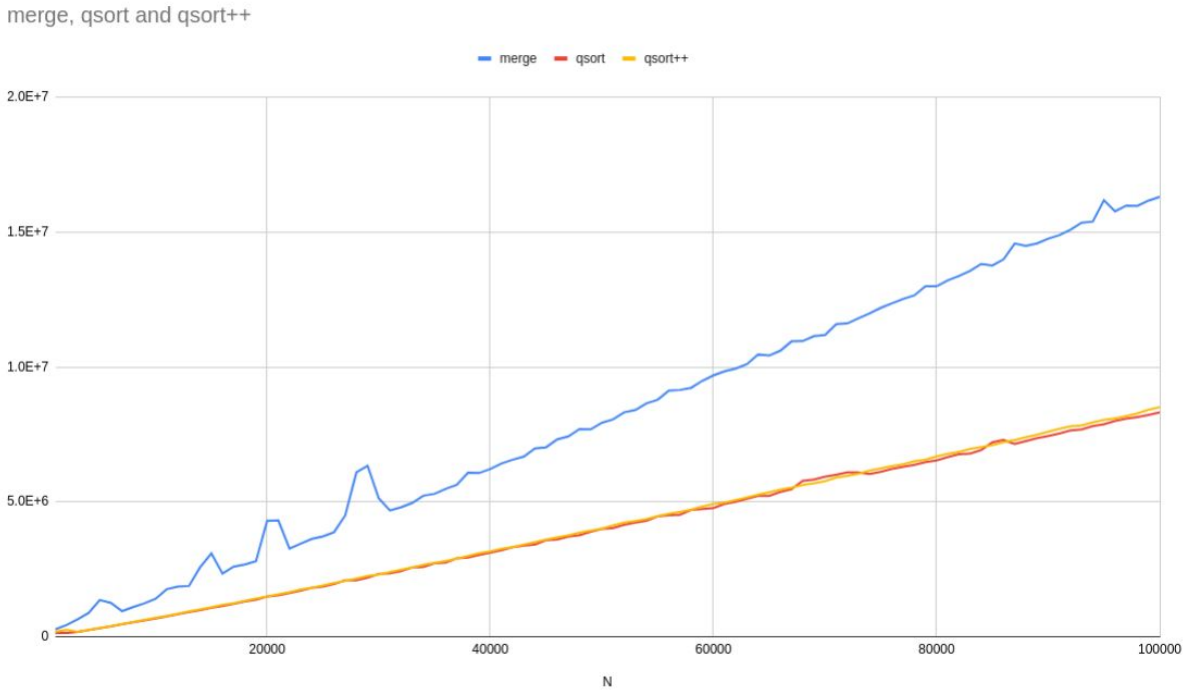
Part 3

Compare the performance of MergeSort, QuickSort and QuickSortEnhanced on a range of inputs (N= 10, 1000, 10000, 100000 etc.) and graph the results of your experiments.

Please find below a graph comparing mergesort (blue), the standard quicksort (red), and the improved quicksort(yellow).



As can be seen from the above graph, quicksort tends to operate significantly faster than mergesort. Though both quicksort and mergesort are O(n logn) functions, quicksort is, in practice, the faster of the two, at least in a single-threaded environment. One interesting feature of the graph is the lack of difference between the improved and the standard mergesort. This might be as a result of the fact that the heuristics introduced to improve efficiency simply weren't parameterised perfectly for the system that the function was tested on. For example, one improvement is the introduction of a call to insertion sort on small sub-arrays, which is helpful to reduce the overall stackframe size, as well as in speeding up the algorithm, This is because, for small values of n (< 10-30, insertion sort will be consistently more efficient that quicksort, because of the overhead that quicksort has in making recursive calls to itself. The value of n for which this is most optimal varies greatly from system to system, and so must be experimentally determined. However, even with an optimal choice for this parameter, there is still no guarantee that it will improve running time, depending on the characteristics of the input.