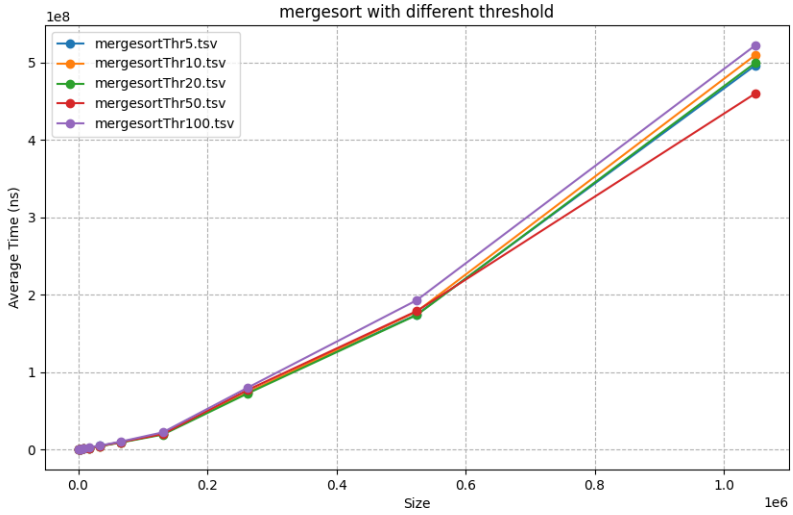
Assignment 4 Analysis Document

Ray Ding

Who are your team members?

Xiyao Xu

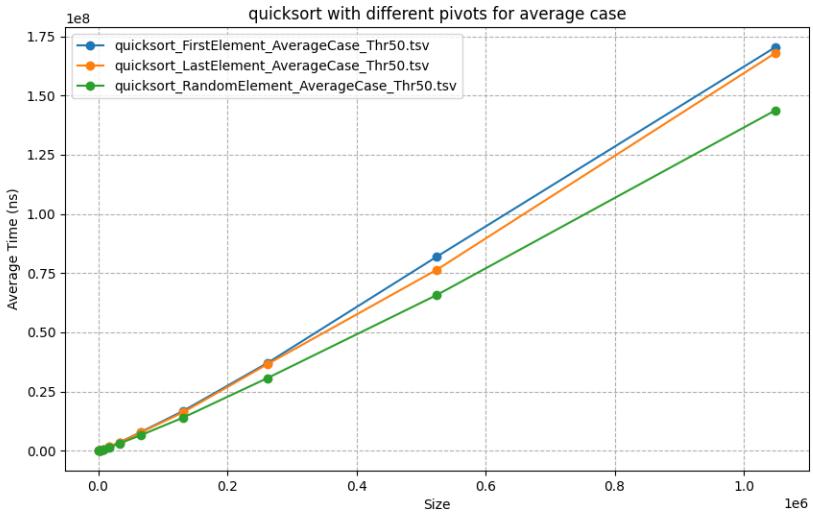
Mergesort Threshold Experiment



As shown by the figure below, the red curve is flattest in the five. Thus, on permuted-order lists, the best threshold value for mergesort switches over to insertion sort is 50.

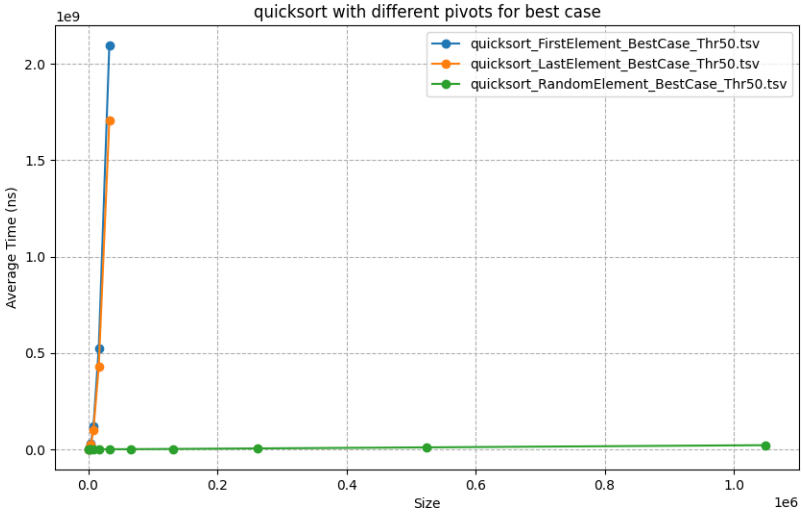
Quicksort Pivot Experiment

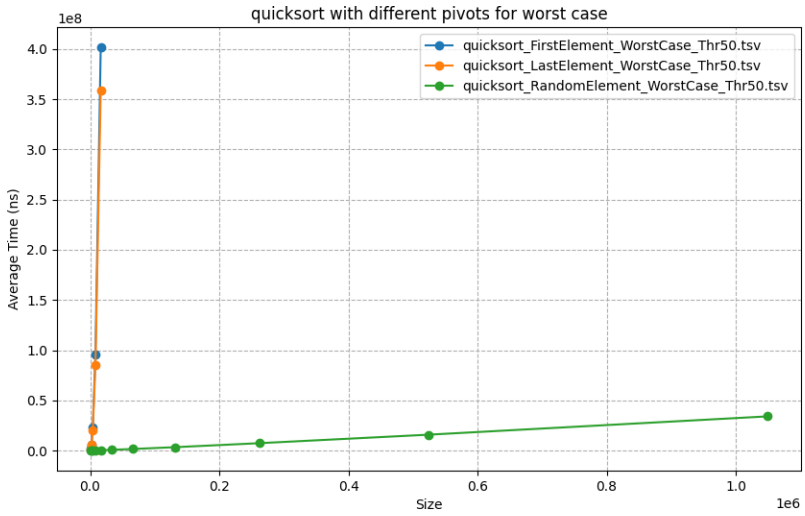
The below graph shows the running time of the quicksort algorithm using three different pivot selection strategies. All three lines show similar trends. The result supports the assertion that QuickSort operates with average-case efficiency when the input data is in a random order, regardless of the pivot selection strategy. It also illustrates the robustness of QuickSort in practical applications when dealing with average-case scenarios.



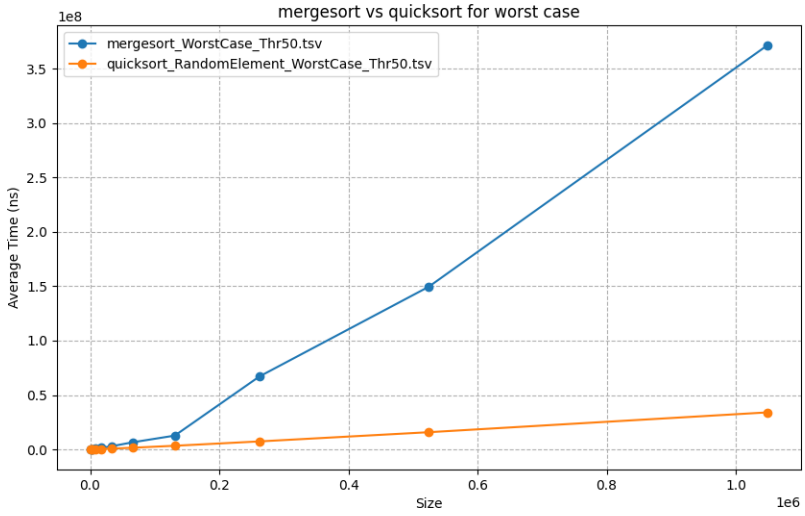
Mergesort vs. Quicksort Experiment

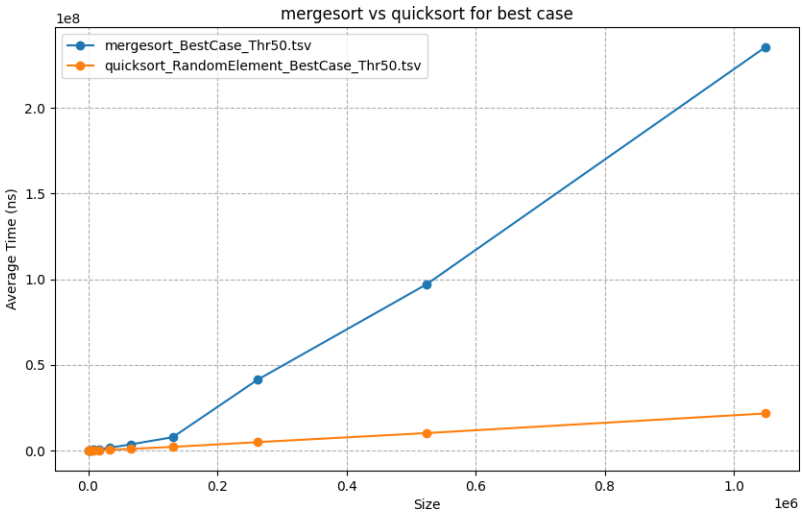
The best pivot strategy for quicksort is to use random element pivot for all three cases. Especially for the best and worst case. The data of first and last elements as pivots spikes sharply and shows characters of O(N2) time complexity, which leads to the worst performance.

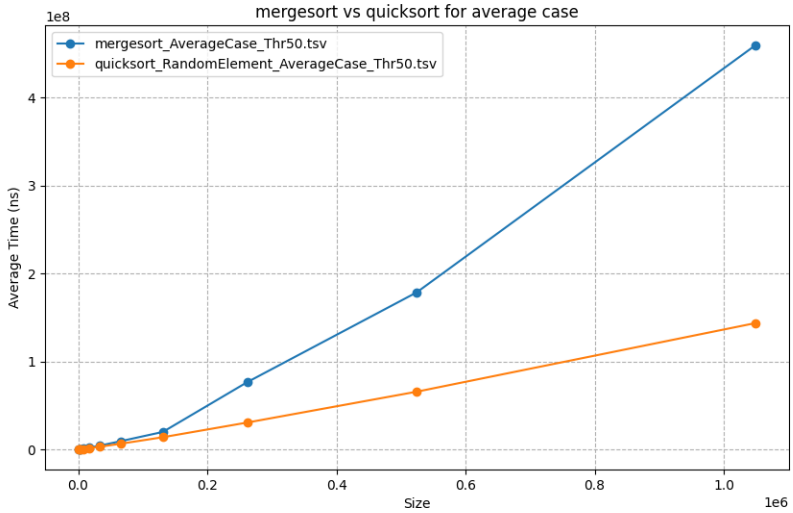




The following three figures display the comparison of mergesort and quicksort algorithms for three cases.







Do the actual running times of your sorting methods exhibit the growth rates you expected to see?

Yes, both mergesort and quicksort exhibits a linearithmic O(NlogN) growth rate for all three cases in running time as the size of input increases, which is consistent with the theoretical performance. Mergesort is known to have a consistent performance regardless of the initial order of elements due to its divide-and-conquer approach. On the other hand, quicksort

demonstrates a much slower growth rate in running time, which suggests that the random pivot selection is a very effective strategy for preventing the algorithm from degrading to quadratic performance.