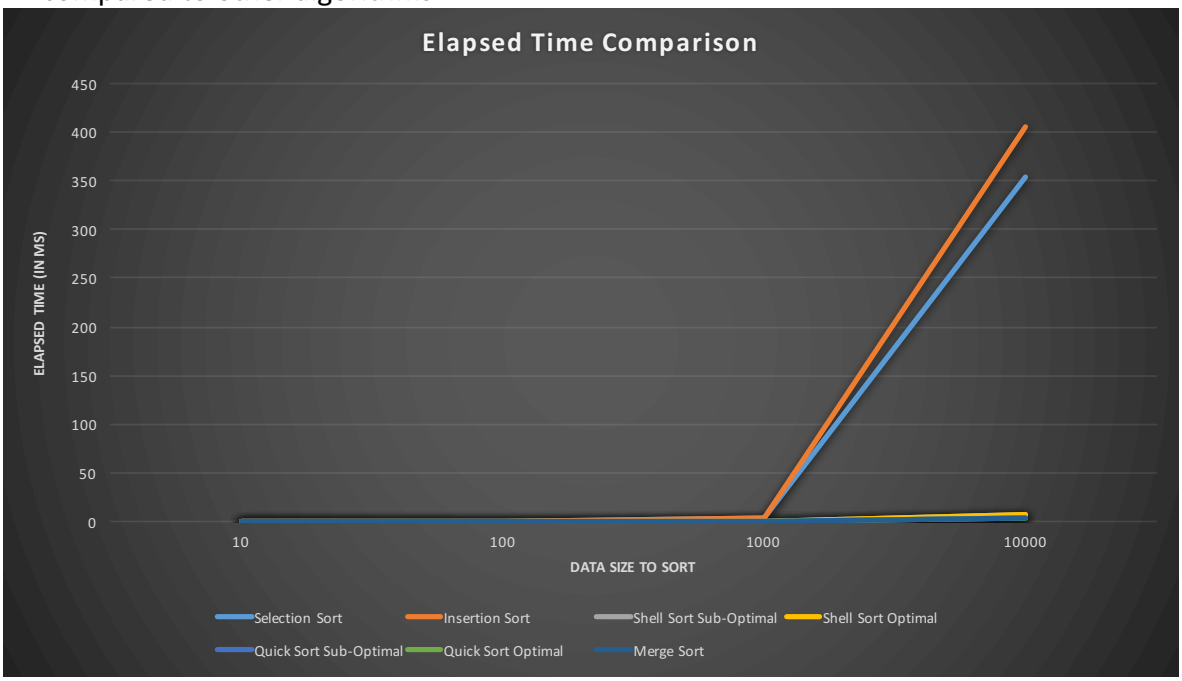
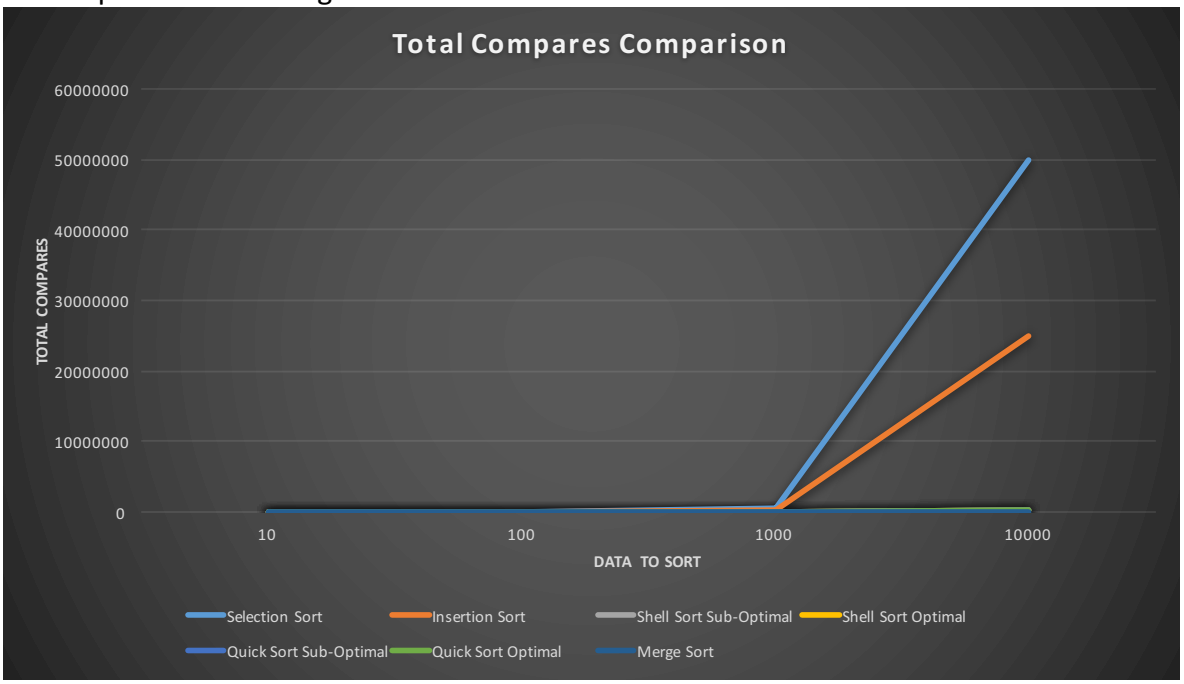
Assignment #12: Sorting Algorithms (Extra credit)

Elapsed Time, Total Moves and Total Compares Comparison of All Sorting Algorithms

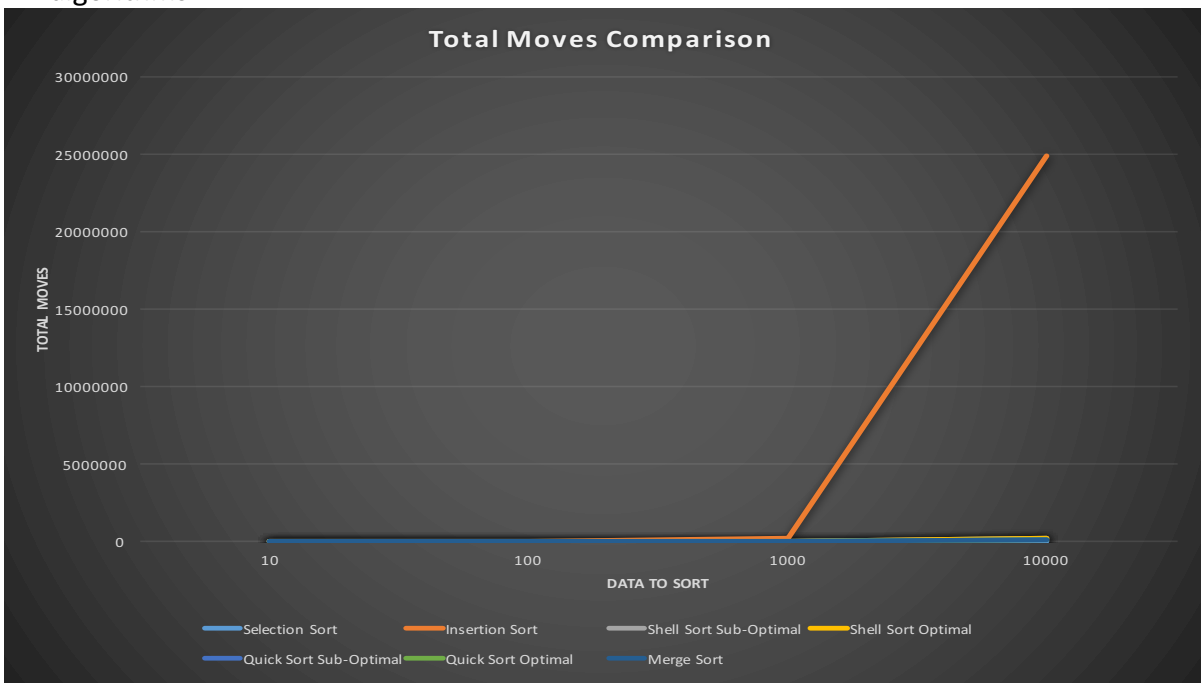
1. Elapsed time increases at a much larger rate for Insertion Sort and Selection Sort as compared to other algorithms



2. Total Compares increase at a much larger rate for Insertion Sort and Selection Sort as compared to other algorithms

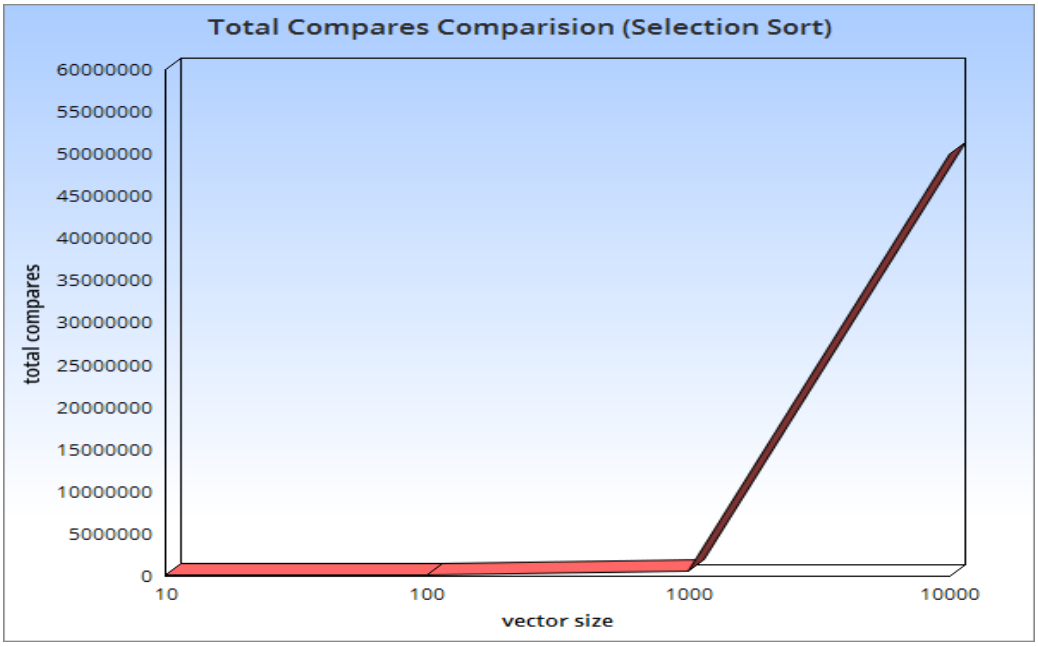


3. Total Moves increase at a much larger rate for Insertion Sort as compared to other algorithms

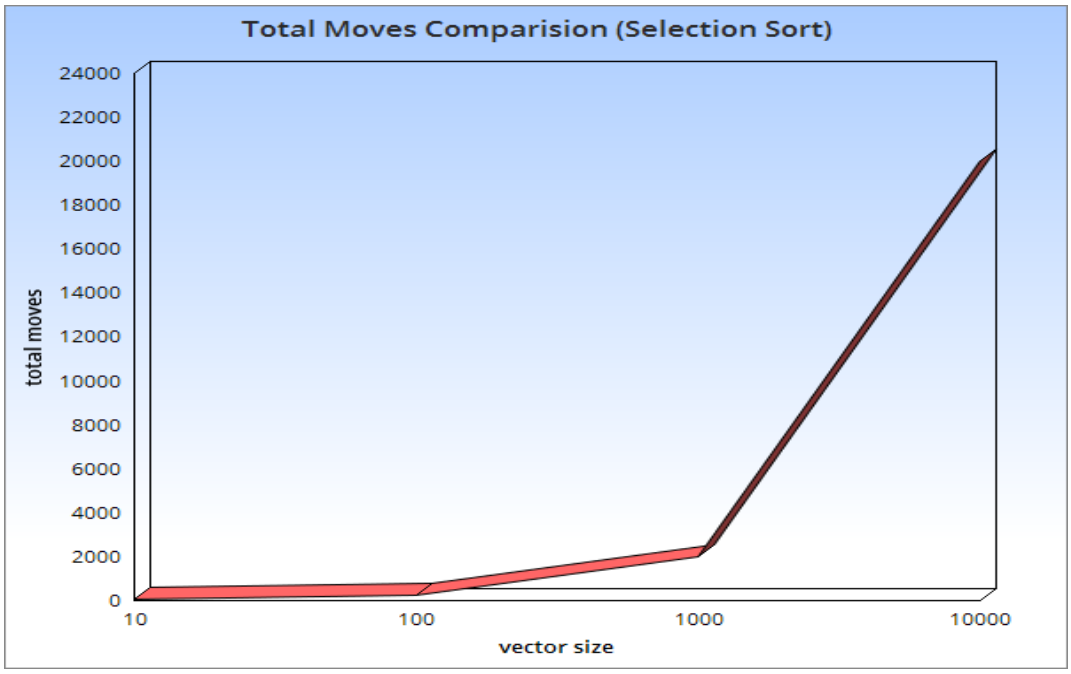


Comparison Charts for Selection Sort

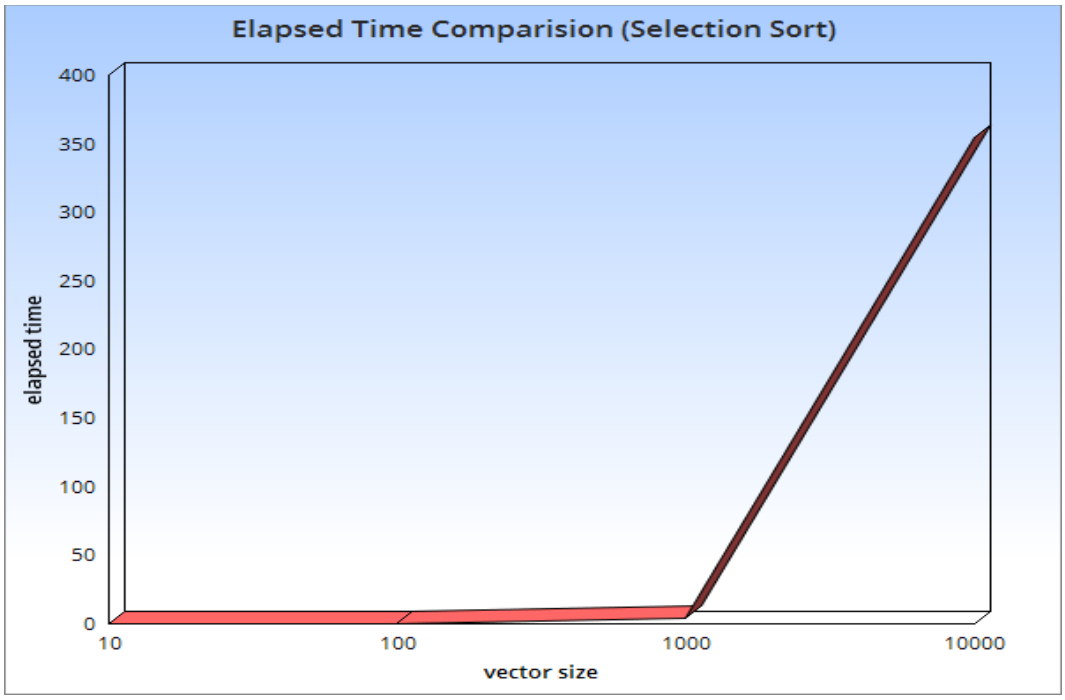
1. Total compares increase exponentially for Selection Sort as the data size to be sorted increases.



2. Total moves increase exponentially for Selection Sort as the data size to be sorted increases.

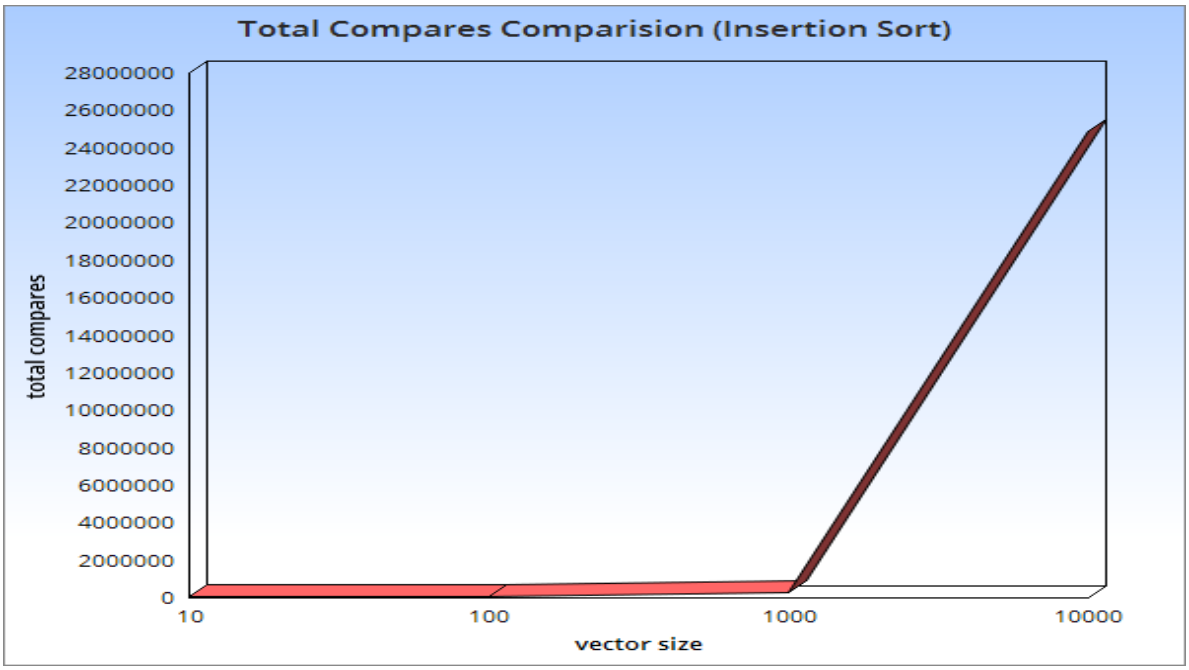


3. Elapsed time increases exponentially for Selection Sort as the data size to be sorted increases.

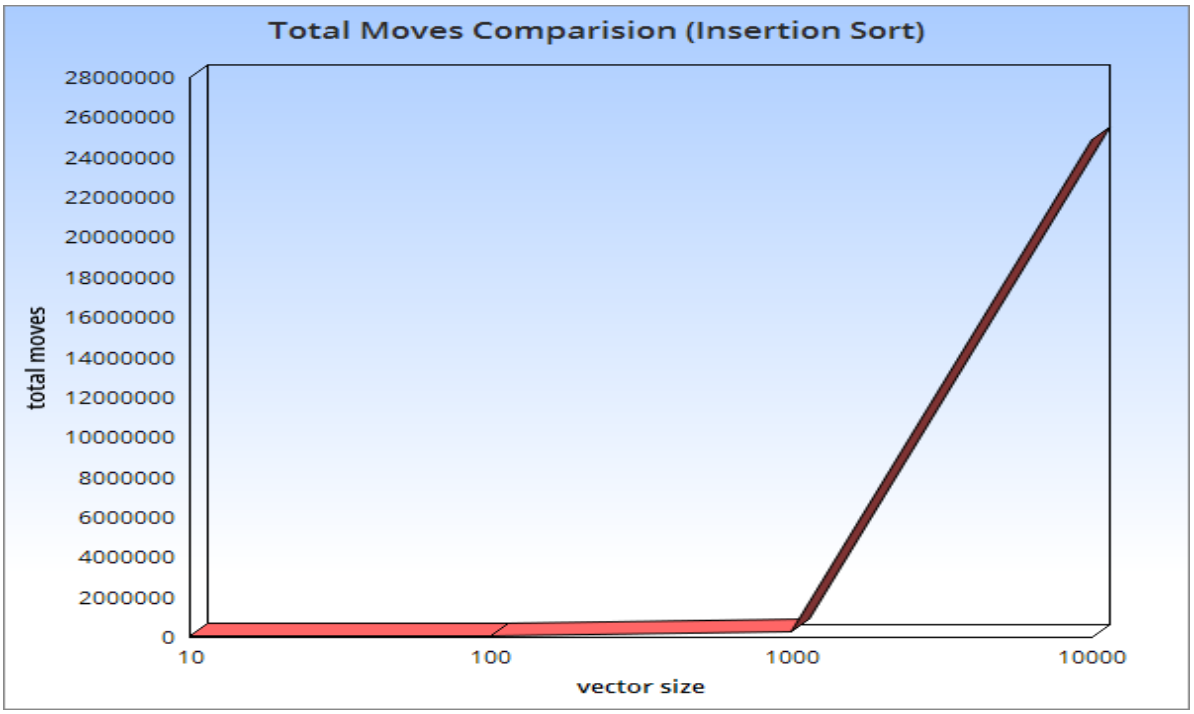


Comparison Charts for Insertion Sort

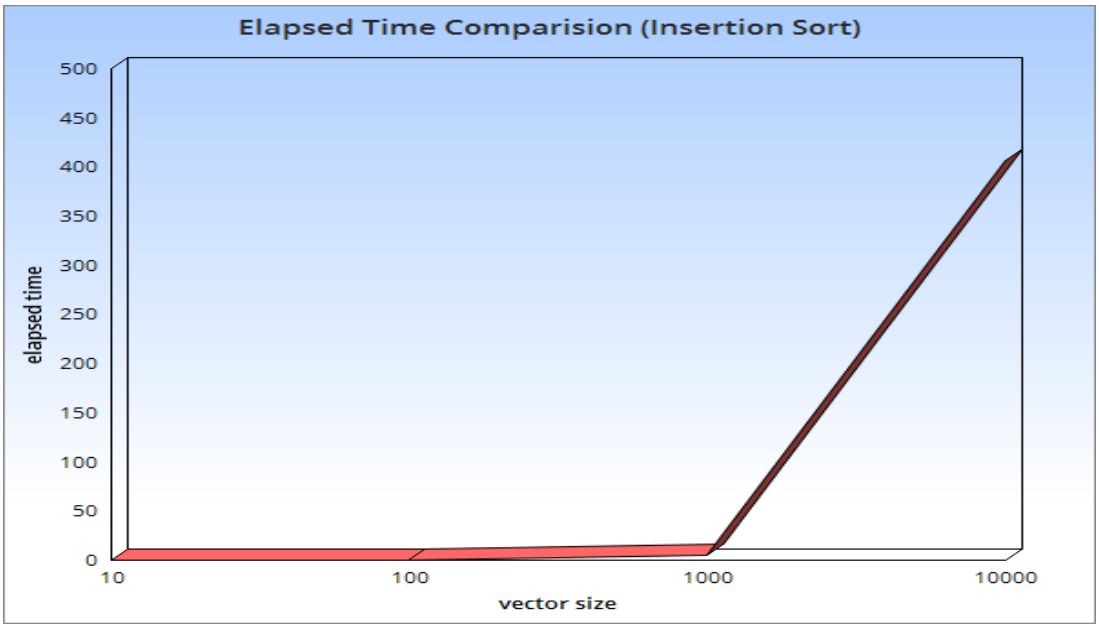
1. Total compares increase exponentially for Insertion Sort as the data size to be sorted increases.



2. Total moves increase exponentially for Insertion Sort as the data size to be sorted increases.

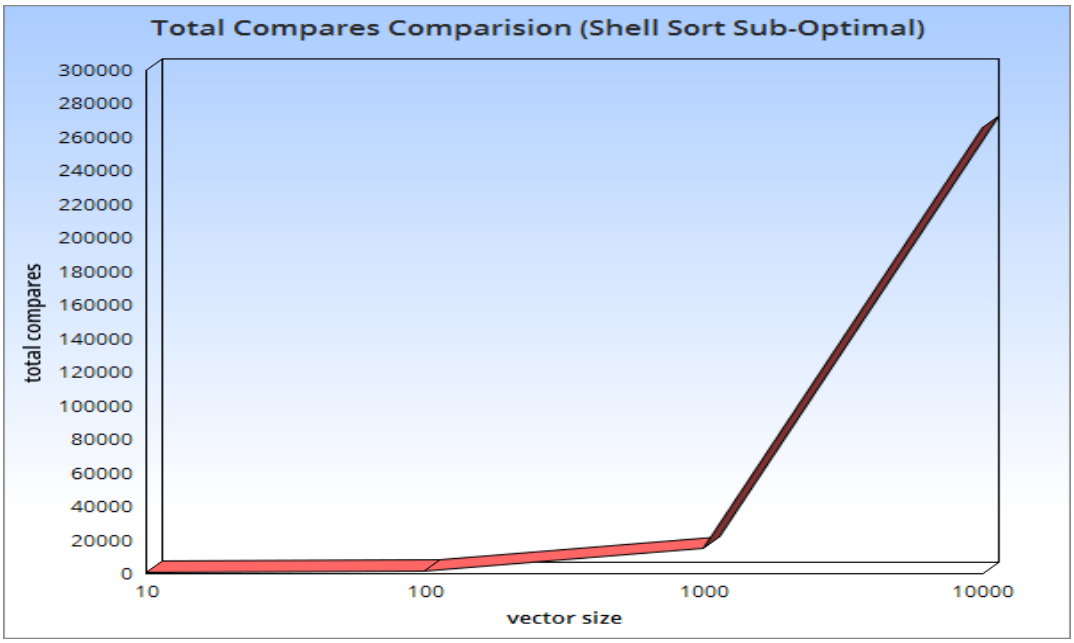


3. Elapsed Time increases exponentially for Insertion Sort as the data size to be sorted increases.

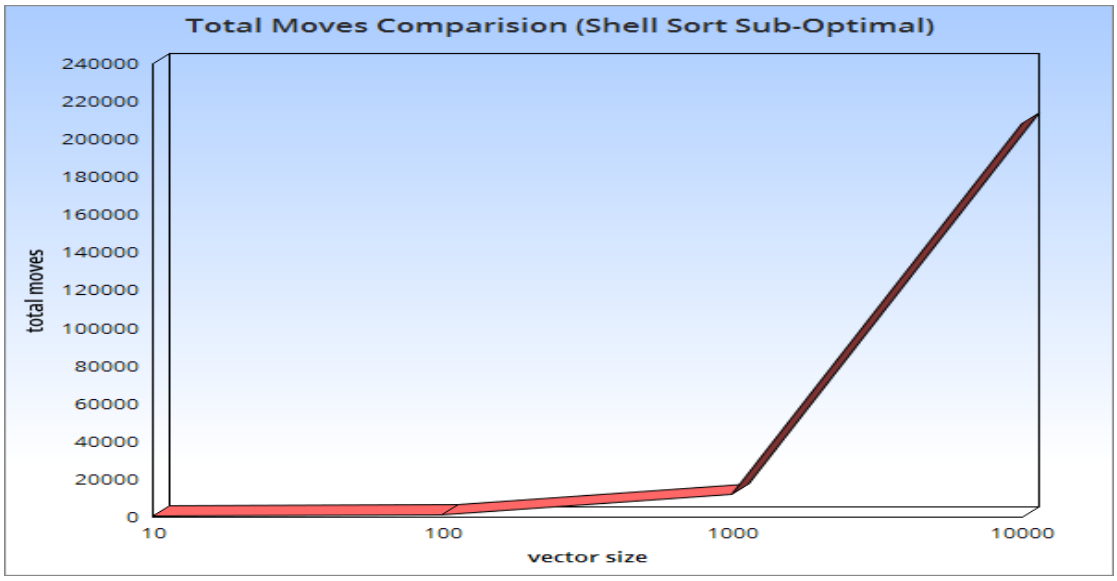


Comparison Charts for Shell Sort Sub-Optimal

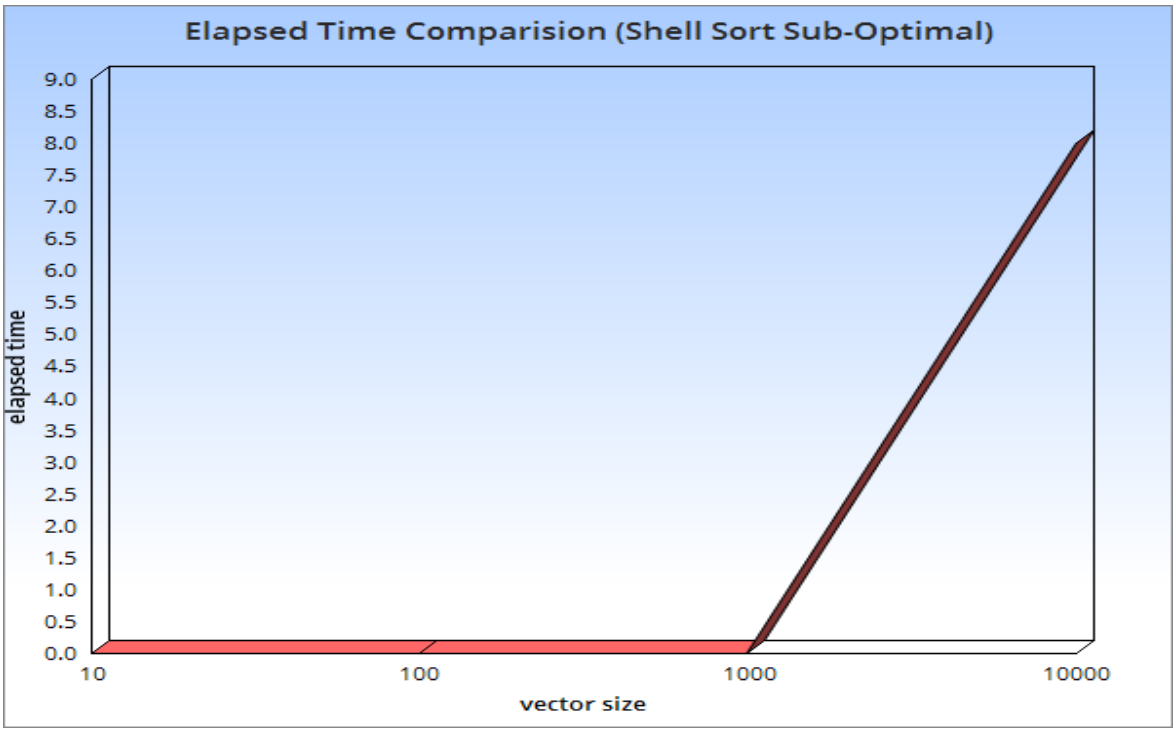
1. Total Compare increases exponentially for Shell Sort Sub-Optimal as the data size to be sorted increases.



2. Total Moves increase exponentially for Shell Sort Sub-Optimal as the data size to be sorted increases.

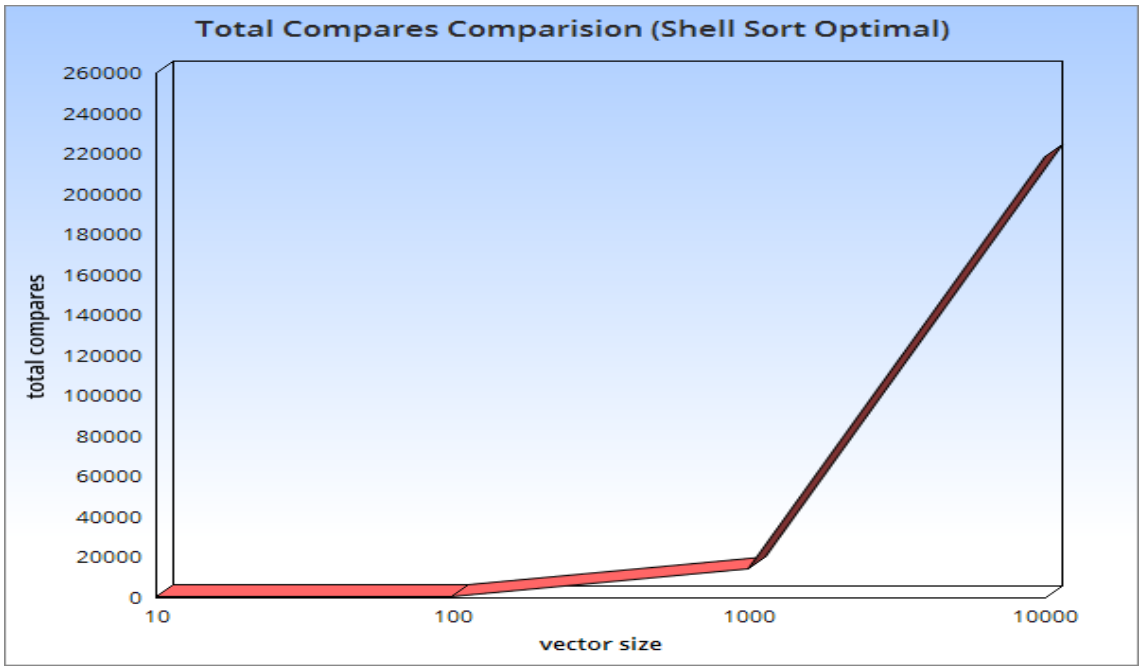


3. Elapsed Time increases exponentially for Shell Sort Sub-Optimal as the data size to be sorted increases.

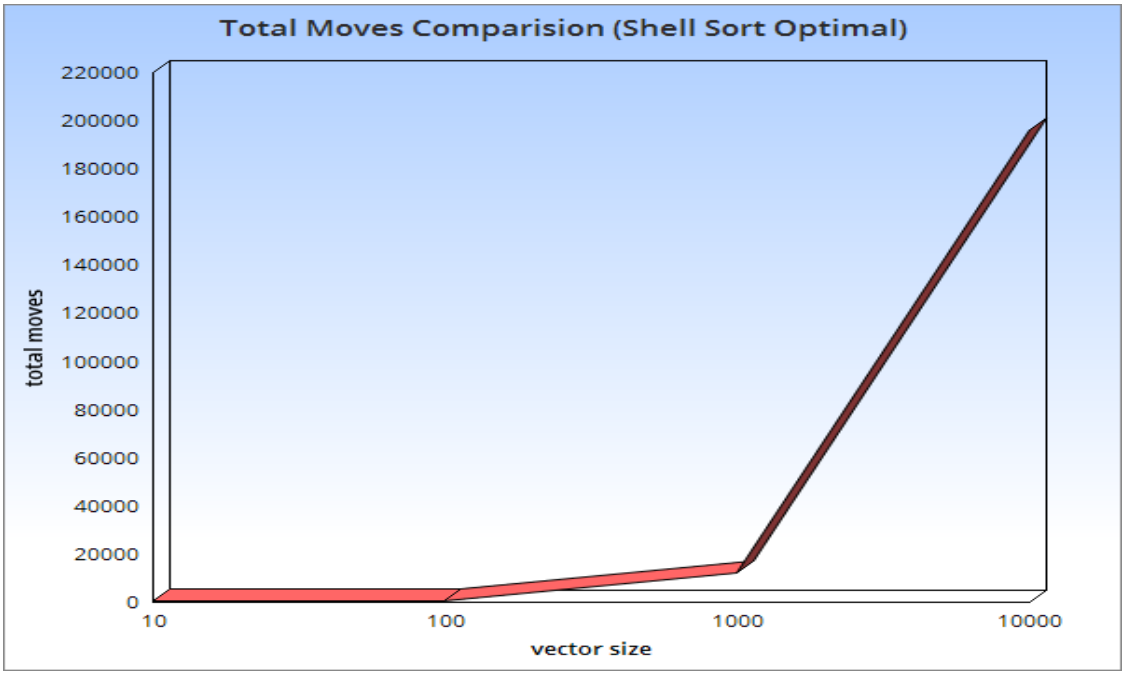


Comparison Charts for Shell Sort Optimal

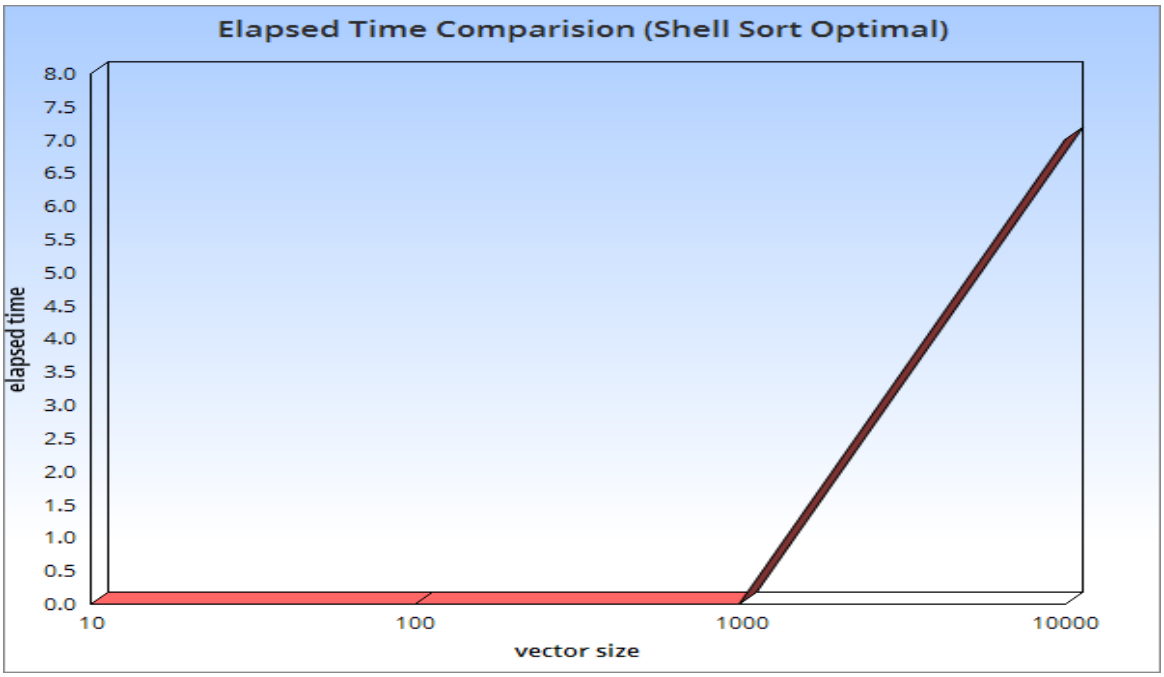
1. Total Compares increase exponentially for Shell Sort Optimal as the data size to be sorted increases.



2. Total Moves increase exponentially for Shell Sort Optimal as the data size to be sorted increases.

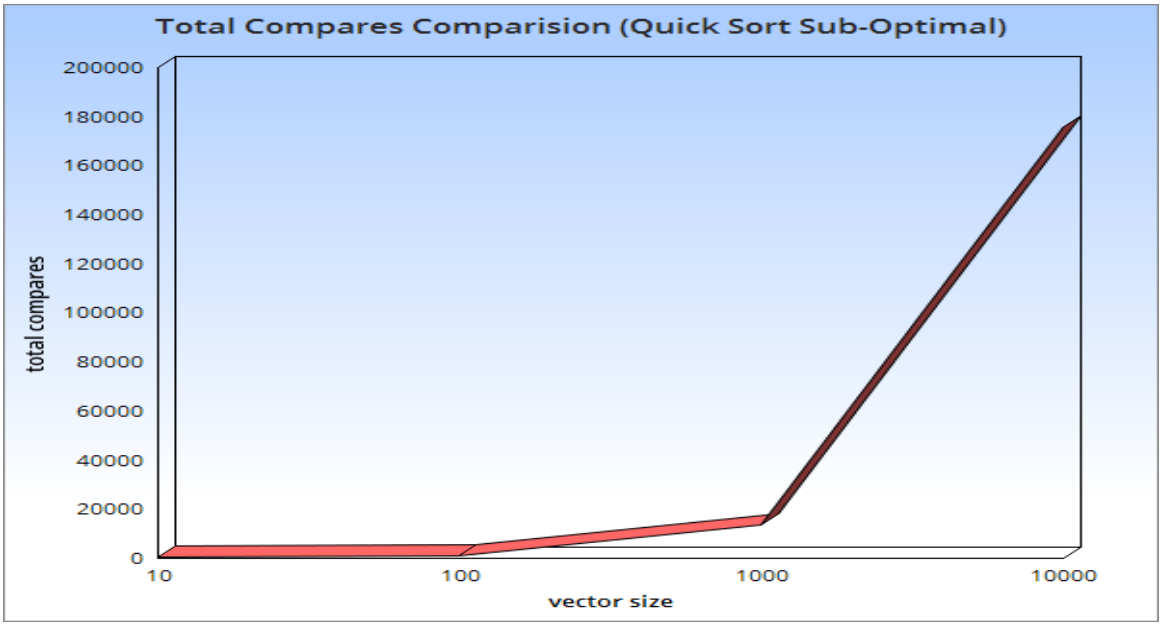


3. Elapsed Time increases exponentially for Shell Sort Optimal as the data size to be sorted increases.

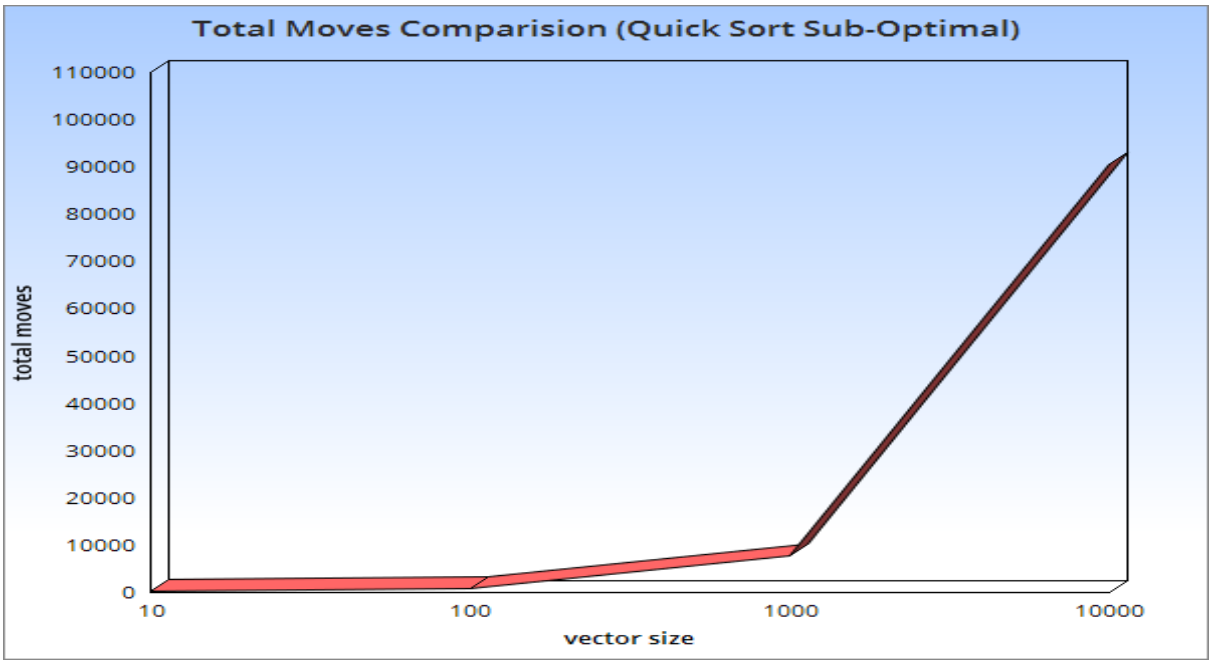


Comparison Charts for Quick Sort Sub-Optimal

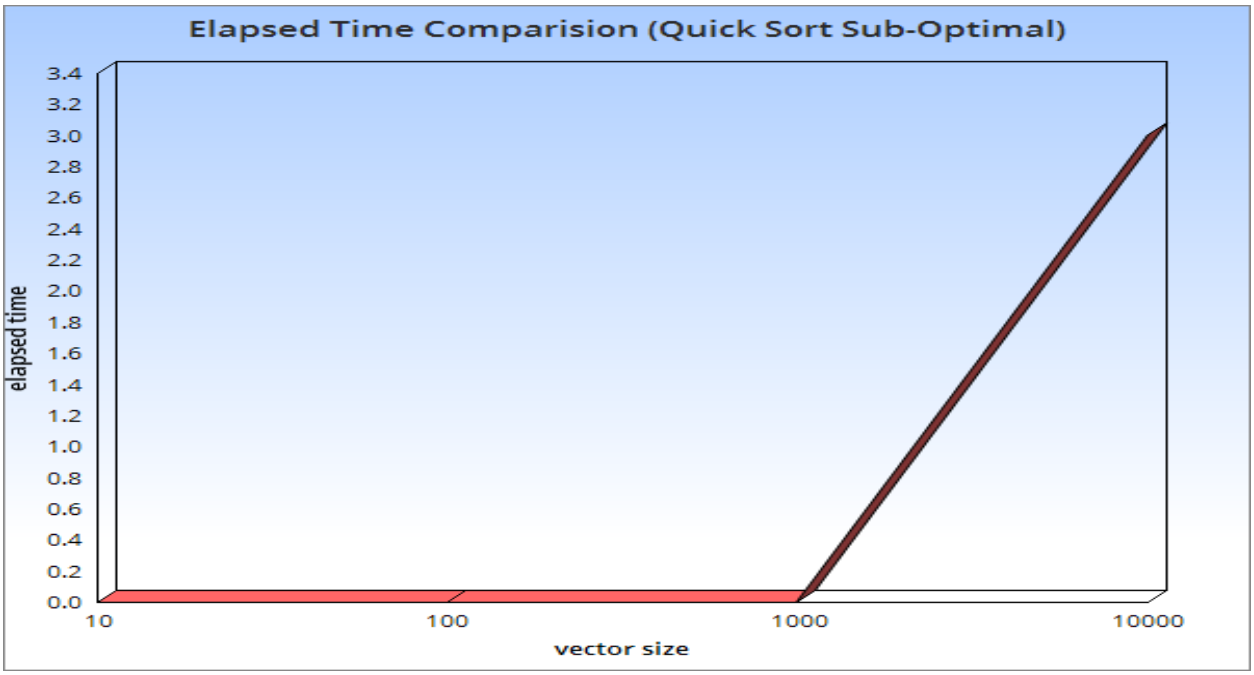
1. Total Compares increase exponentially for Quick Sort Sub-Optimal as the data size to be sorted increases.



2. Total Moves increase exponentially for Quick Sort Sub-Optimal as the data size to be sorted increases.

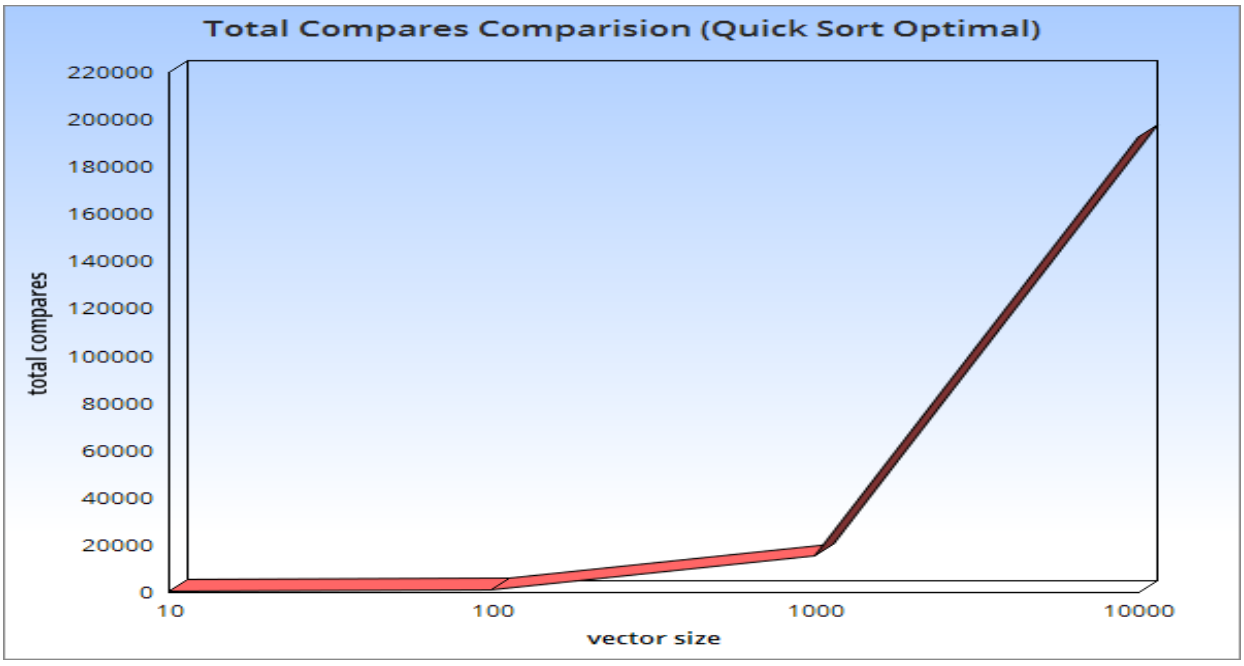


3. Elapsed Time increases exponentially for Quick Sort Sub-Optimal as the data size to be sorted increases.

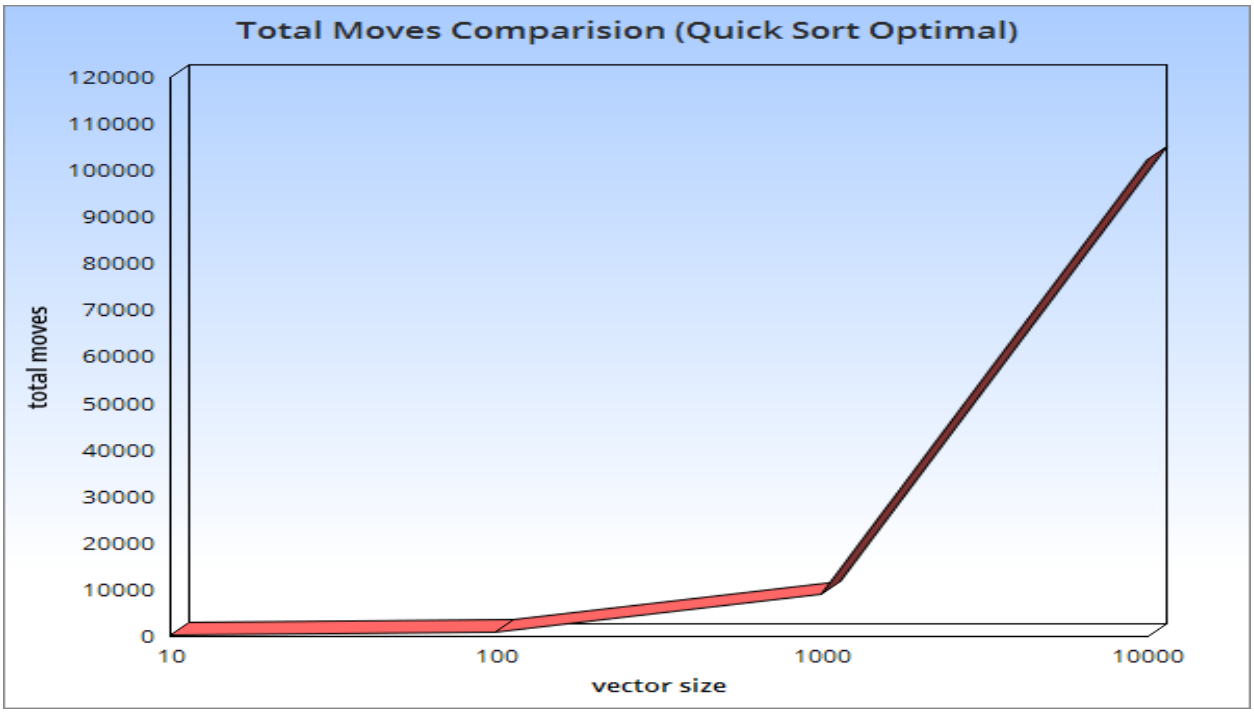


Comparison Charts for Quick Sort Optimal

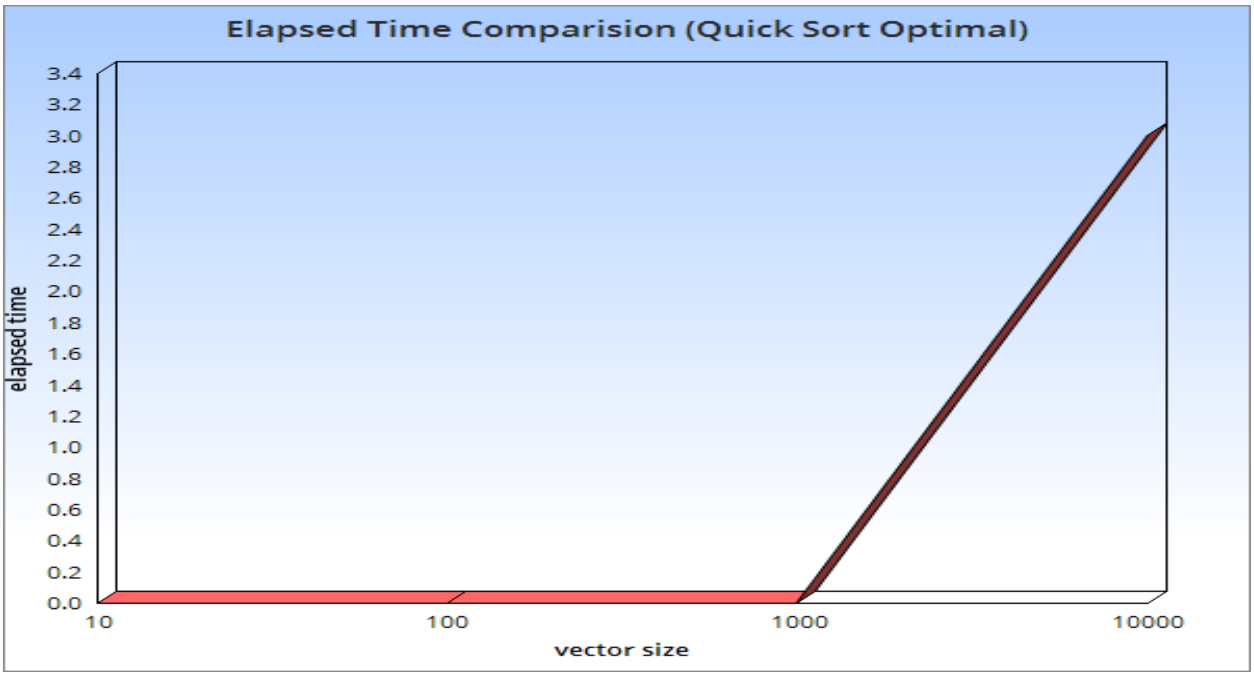
1. Total Compares increase exponentially for Quick Sort Optimal as the data size to be sorted increases.



2. Total Moves increase exponentially for Quick Sort Optimal as the data size to be sorted increases.

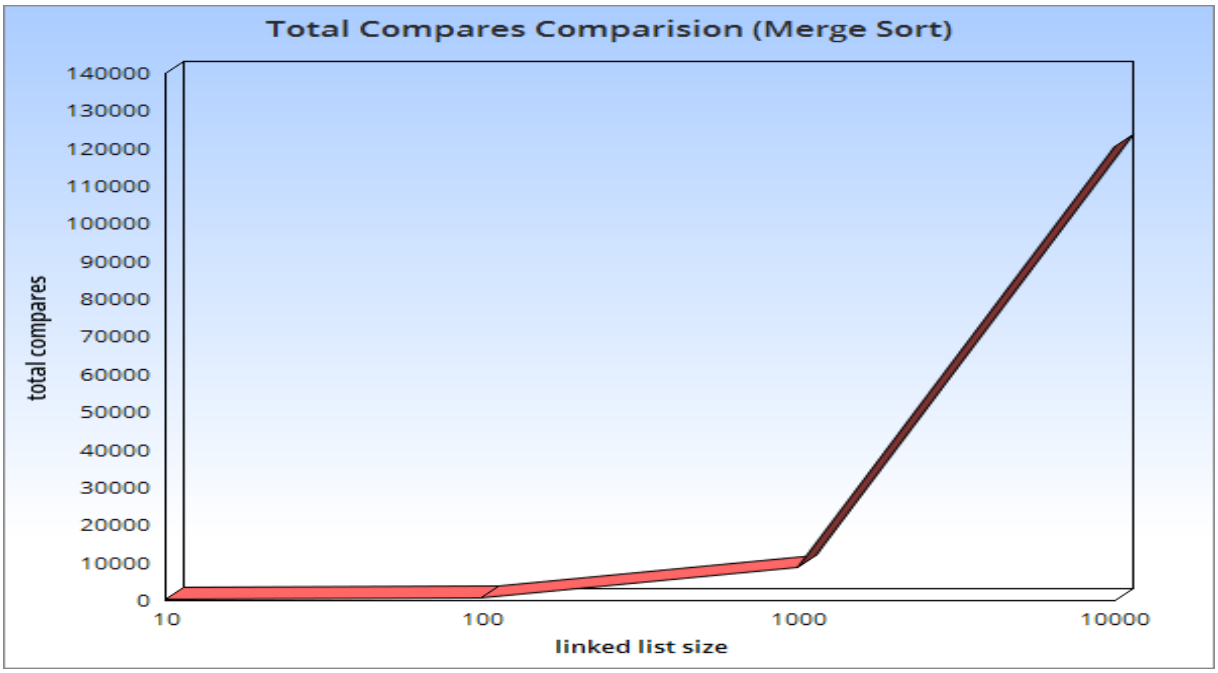


3. Elapsed Time increases exponentially for Quick Sort Optimal as the data size to be sorted increases.

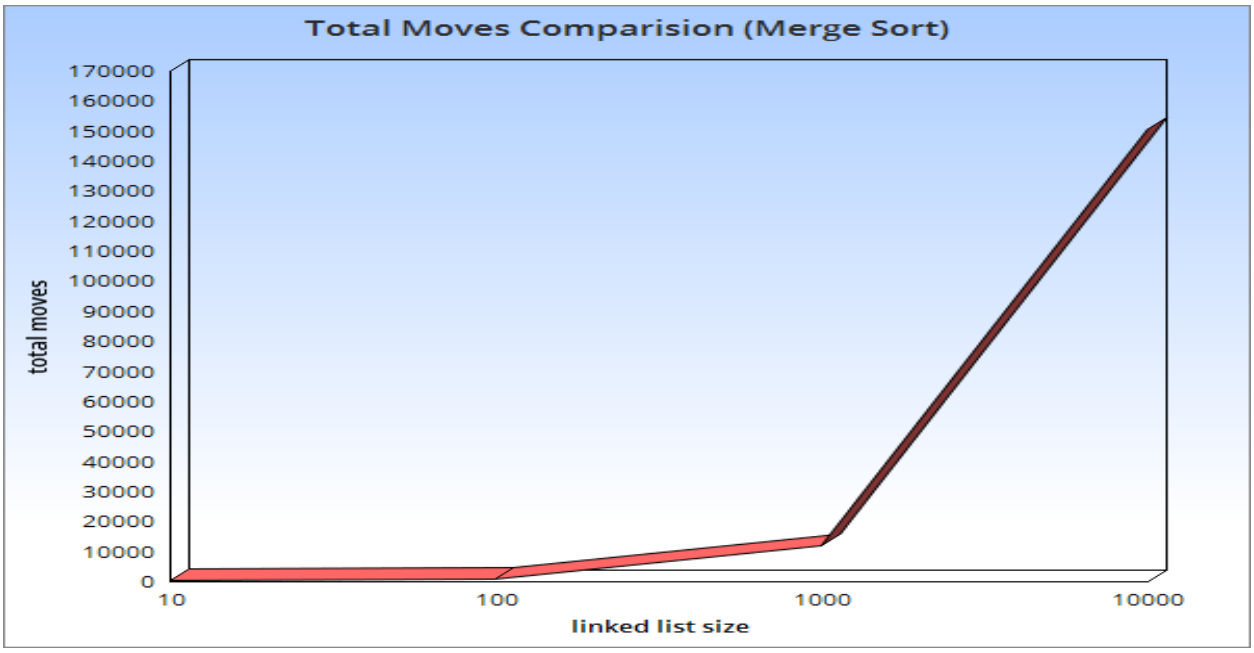


Comparison Charts for Merge Sort

1. Total Compares increase exponentially for Merge Sort as the data size to be sorted increases.



2. Total Moves increase exponentially for Merge Sort as the data size to be sorted increases.



3. Elapsed Time increases exponentially for Merge Sort as the data size to be sorted increases.