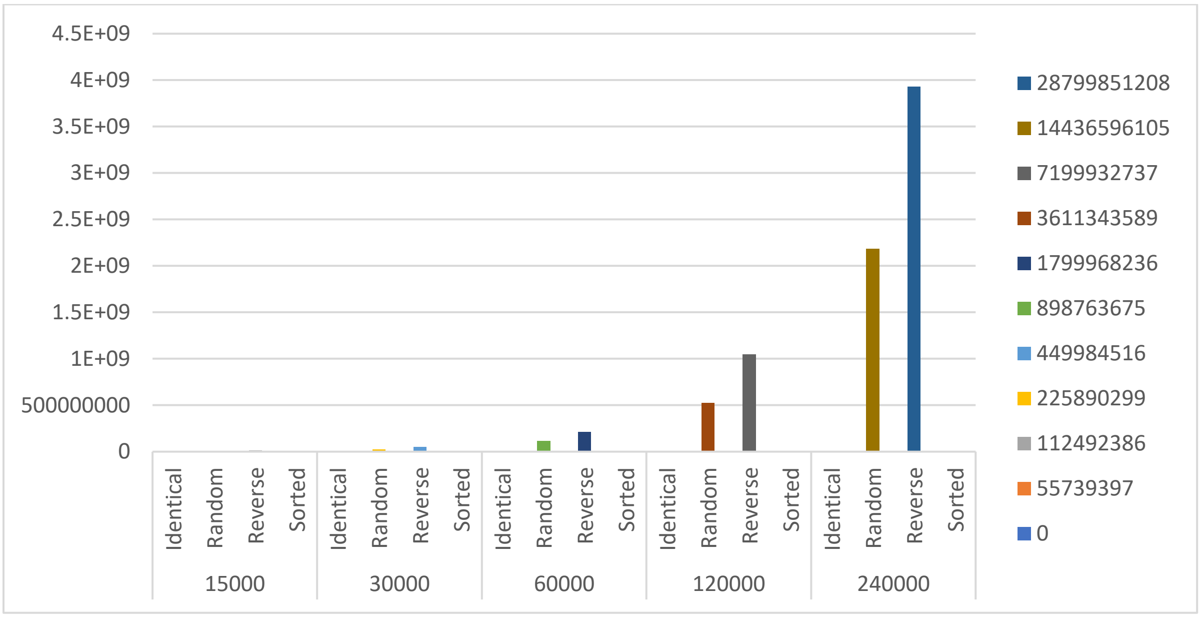
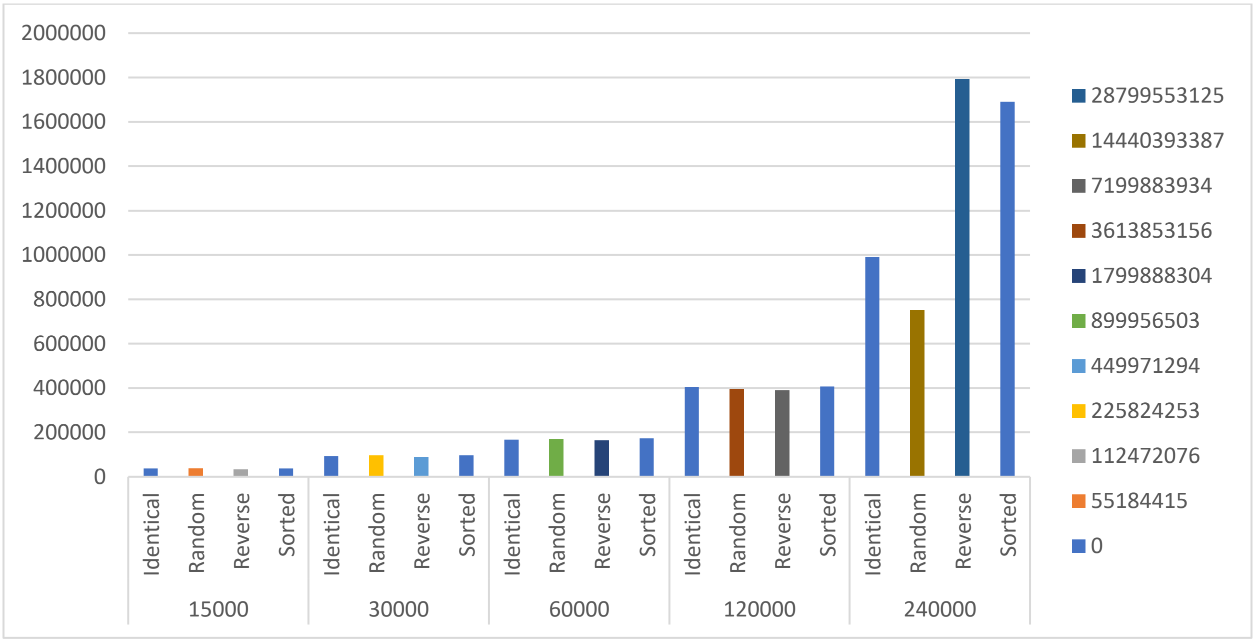
## Mehrdad Yadollahi (0020481623)

## Theoretical Analysis:

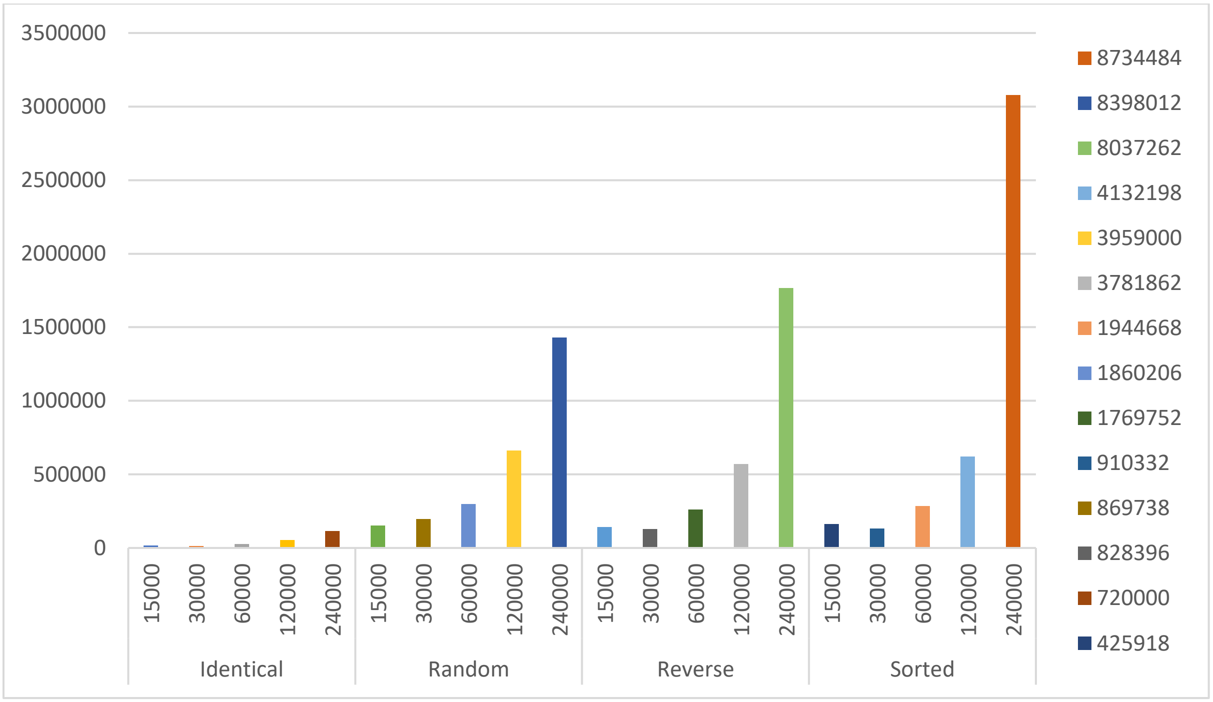
1. **Insertion Sort**: The worst-case time complexity of insertion sort is ϴ(n^2), where n is the number of elements in the list. The best-case time complexity is ϴ(n) when the list is already sorted, and it is a linear function.



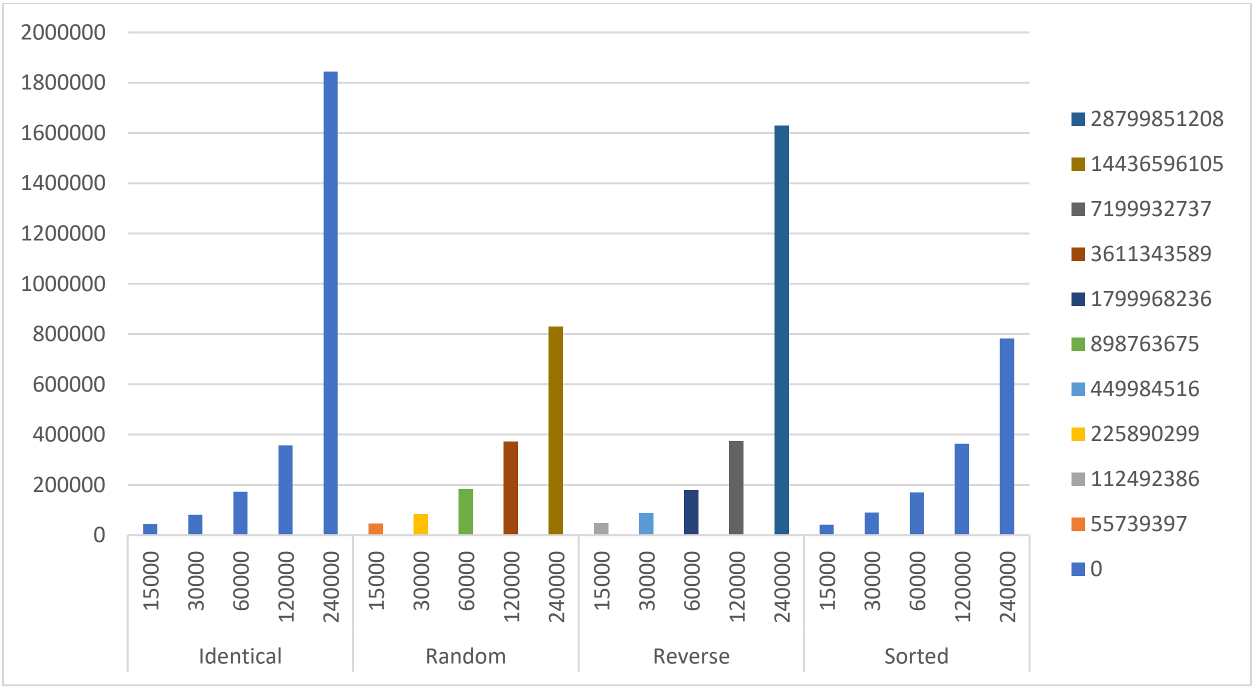
2. **Merge Sort**: The time complexity of merge sort is ϴ(n log n) in all cases (worst-case, average-case, and best-case). The space complexity is ϴ(n) due to the need for additional memory for the merge step.



3. **Heap Sort**: The time complexity of heap sort is Θ(n log n) in all cases and it is in-place like the insertion sort.



4. **4-way Merge Sort**: This is another version of merge sort. It uses a 4-way Merge, which could potentially improve performance for very large lists. The time complexity is still Θ(n log n), but it might have a smaller constant factor due to the 4-way merge.



## Analysis and discussion:

Each sorting algorithm has its own strengths and weaknesses, and the choice of which one to use depends on various factors, including the size and nature of the data to be sorted. For small datasets or nearly sorted data, Insertion Sort may be a good choice. Merge Sort is a reliable general-purpose algorithm, while Heap Sort is efficient for larger datasets and doesn't require additional space. 4-Way Merge Sort can potentially provide advantages for very large datasets, but it introduces additional complexity.