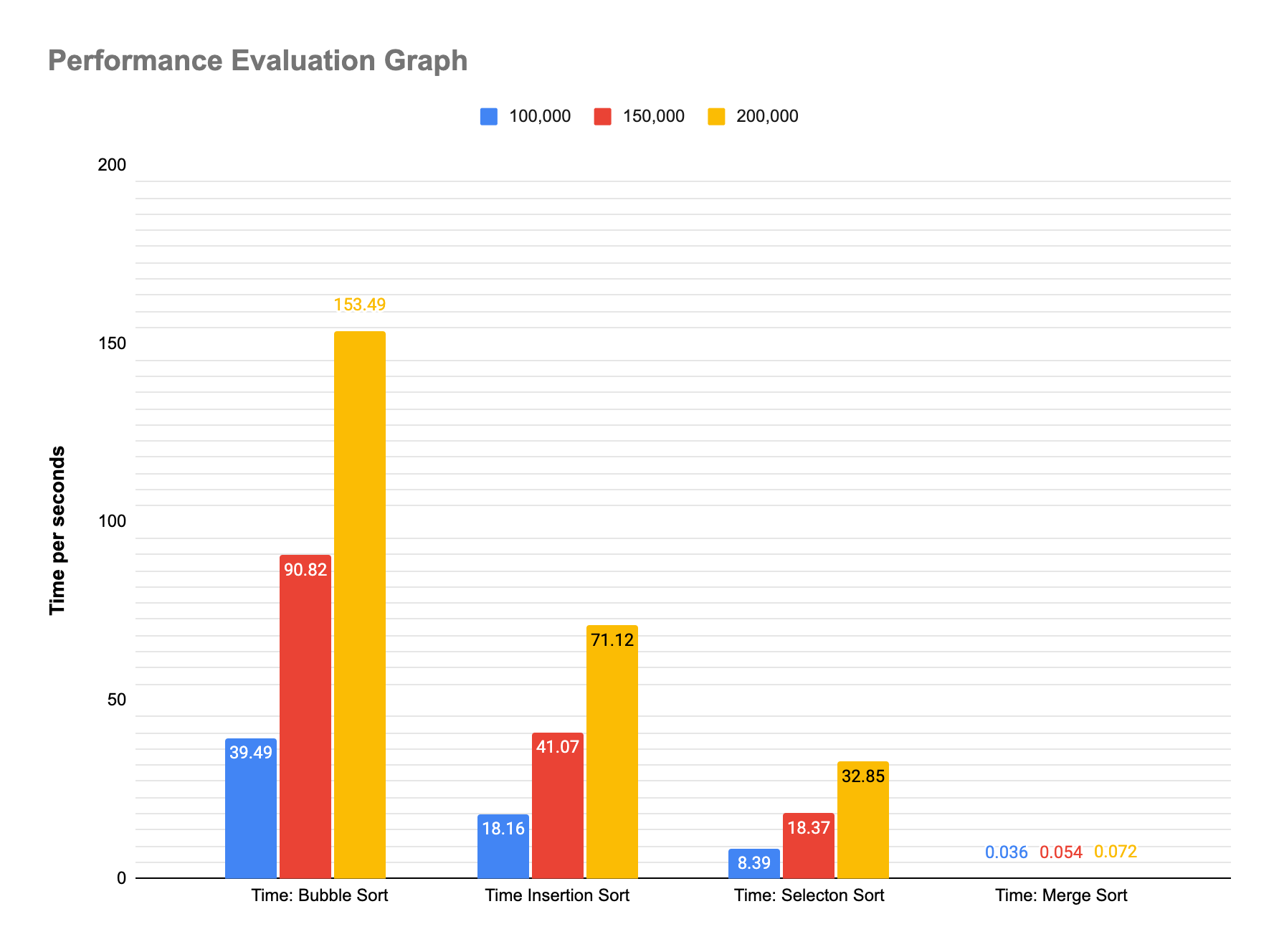
Performance Comparison Table

| **Dataset Size** | **Time**  **Bubble Sort**  **(Second)** | **No. of Swapping for Bubble Sort** | **Time**  **Insertion Sort**  **(Second)** | **No. of Swapping for Insertion Sort** | **Time**  **Selection Sort**  **(Second)** | **No. of Swapping for Selection Sort** | **Time**  **Merge Sort**  **(Second)** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 100,000 | 39.49 | 2496277556 | 18.16 | 2496277556 | 8.39 | 99898 | 0.036 |
| 150,000 | 90.82 | 5609229756 | 41.07 | 5609229756 | 18.37 | 149851 | 0.054 |
| 200,000 | 153.49 | 9983487955 | 71.12 | 9983487955 | 32.85 | 199795 | 0.072 |



**Conclusion:** In conclusion, when it comes to time and swapping comparison, bubble sort, insertion sort, selection sort, and merge sort all have their own advantages and disadvantages.

Bubble sort is the simplest sorting algorithm among these four, but it has the worst time complexity of O(n^2), making it inefficient for large data sets. It also requires a large number of swaps to sort the array, which makes it less preferable.

Insertion sort has a better time complexity of O(n^2) for average cases but performs better than bubble sort in practice. It also requires fewer swaps than bubble sort. However, it is still not the best choice for large data sets.

Selection sort has the same time complexity as bubble and insertion sort, but it makes fewer swaps. However, its worst-case performance remains the same, which makes it less preferred for large data sets.

Merge sort is a divide-and-conquer algorithm that has a time complexity of O(n log n), making it the most efficient algorithm among these four. It also requires fewer swaps than bubble and insertion sort. However, it requires additional memory space, which may not be suitable for small systems with limited memory.

In summary, when it comes to time and swapping comparison, merge sort is the most efficient algorithm among these four. If we need to sort a small data set with a limited memory space, we can use insertion sort or selection sort. On the other hand, bubble sort should be avoided for larger data sets, as it is inefficient and requires a large number of swaps.