**Analysis of various algorithms—**

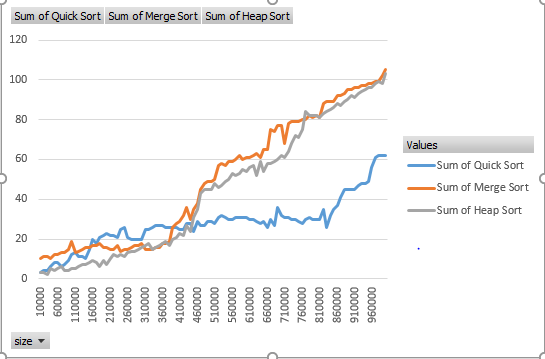
Max data size – 500,000 (for insertion and selection sort)

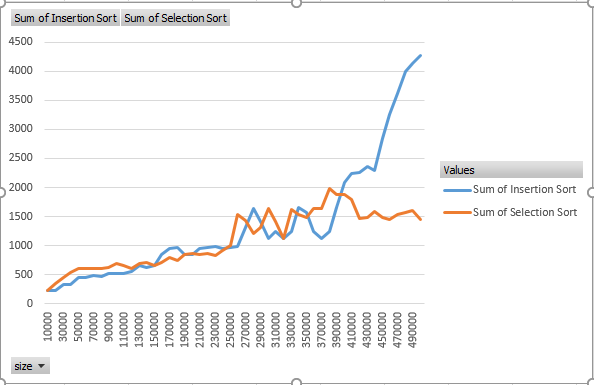
Max data size – 1000000(for merge sort, quick sort, heap sort)

Min data size – 10000



For Partially Sorted Arrays





For Sorted Arrays

For Unsorted Arrays

Complexities-

|  |  |  |  |
| --- | --- | --- | --- |
| Algorithm | Best Case | Worst case | Average case |
| Merge Sort | O(nlogn) | O(nlogn) | O(nlogn) |
| Heap Sort | O(nlogn) | O(nlogn) | O(nlogn) |
| Quick Sort | O(nlogn) | O(n2) | O(nlogn) |
| Selection Sort | O(n2) | O(n2) | O(n2) |
| Insertion Sort | O(n) | O(n2) | O(n2) |

**Observation-**

1. Insertion sort works best on sorted arrays.
2. It can be seen from the graph that the time complexity for insertion sort in case of fully sorted arrays is O(1).
3. Quick Sort and Heap sort are better than merge sort.
4. These algorithms work well with large data sets as large as 1000000.
5. Completely sorted array works as the worst case scenario for quick sort.
6. Selection sort is slow even in case of sorted arrays. The complexity for selection sort is O(n^2).

It iterates through the entire array.

1. Insertion and selection sort do not work well with large arrays as was observed due to which the array size was reduced to 500000.
2. Heap sort works well with large data sets as well.
3. Merge sort gradually reduces speed as the array size keeps on increasing.
4. Although quick sort works fine with small array sizes, it has been observed that for small arrays, insertion sort and selection sort works well.
5. To achieve optimal performance, we can combine two sorting algorithms such as insertion sort and quick sort.
6. For combining quick sort and insertion sort, various values were tried as the threshold.

It was found that if the threshold is kept as high as 1000000, the program executed in 7 minutes for arrays of size from 1000 to 1000000.

However, if the threshold is reduced to a smaller value then the combined sorting technique achieves optimal performance.