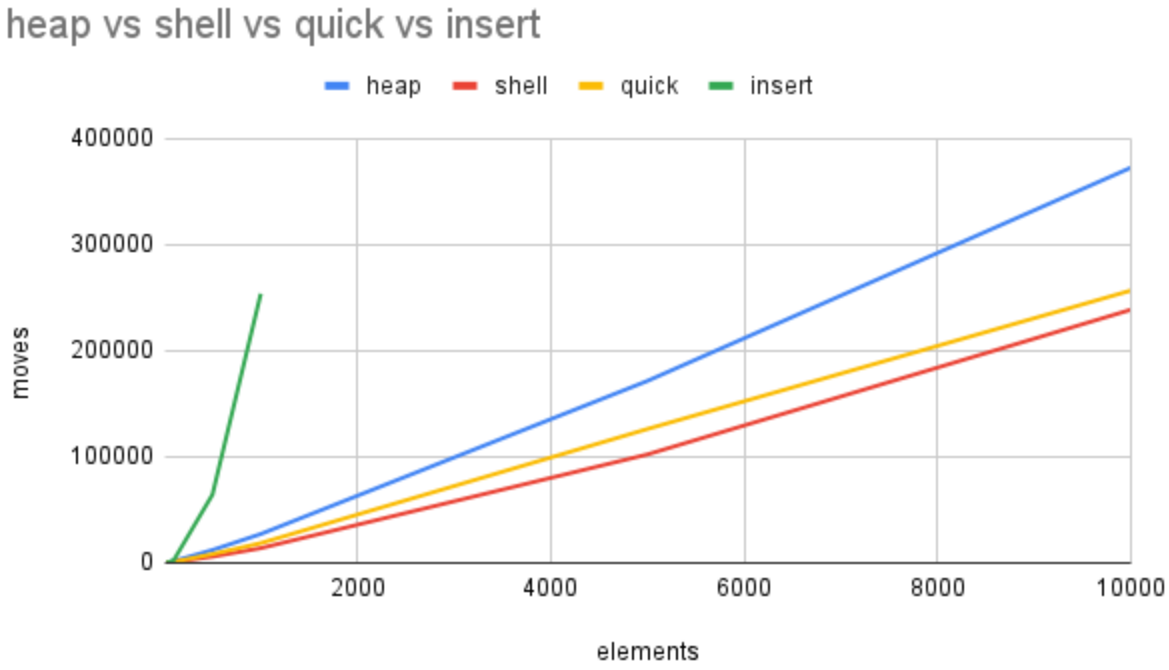

Heap sort:
With heap sort, it appears that it requires more moves when there are more elements, which means that it will take longer to sort when the array is too big. Our heap sort has the parent nod greater or equal to its child's value and then moved to the back, moving the children to the front. When moving, the heap gets fixed.

Shell sort:
Shell sort is a variation of insertion sort, and sorts pairs of elements that are gaps apart. These gaps will determine the time complexity of the shell sort. Bigger gaps will result in longer time sorts. In our case, it looks to be the best, with the best moves to elements ratio.

Quick sort:
Quick sort is sorted by choosing an element as a pivot, and elements less than the pivot go to the left of it, while the elements larger than the pivot go to the right. In our case, quick sort is worse than shell, but better than heap.

Insertion sort:
Insertion sort looks to be the worse sort when it comes to dealing with large numbers of elements in the array. It is probably because of it dealing with one element at a time. With more elements more comparisons happen, which increases the moves.