Karim Elshabassy

Standard sort	0,20,378
Half selection	24,5622,47178797+
Half heap	0,6,251
In place merge	1,46,656
merge	2,93,3263
quick	0,6,83
Worse case quick	408,970,859

From this chart we can see how long each sort takes for 1k,31k, and 1 million, something interesting is that standard sort was faster than both merge sorts and quick sort completed the 1 million sizes input way faster than the others, but when it came to the worse case there was not that big of a difference between the 3 sizes. I also found it unexpecting that half-sort took so long that I got an error that it took too long to run

Standard sorts time complexity is usually O(n log n), where n is the size of the vector and space complexity of O(1), because of how it is in pace.

Selection sort has a time complexity of O(n^2), where n is the size of the vector and its space complexity is O(1) since it performs in-place swaps.

Half heap has a time complexity of O(n log n), where n is the size of the vector. Building the heap has a time complexity of O(n), and the repeated removal of the minimum has a time complexity of O(n log n). The space complexity is O(1) since it performs in-place swaps and uses a constant amount of space.

In-place merge has a time complexity of O(n log n), where n is the size of the vector. The space complexity is O(1) since it performs in-place sorting. This is because of the comparisons and recursive calls, on average it performs n log n / 2 comparisons.

Merge has a time complexity of O(n log n), where n is the size of the vector. The space complexity is O(n) due to creation of temporary vectors during the merge process. Same average comparisons as the in-place version.

Finally Quick Select has a time complexity of O(n), where n is the size of the input vector.
The worst-case time complexity is O(n^2), but this is unlikely to occur with the median-of-three pivot selection and the insertion sort cutoff for small partitions this however changes when we use the worst case vector because of the multiplied amount of calls. The space complexity is O(log n) due to the recursive call stack.