Research and implement two "QuickSort" algorithms: with and without recursion.

Let's say we have two arrays - static and dynamic ones with the following sizes: 100, 1000, 100000, 1000000, 1000000000. Using the both of the arrays, investigate the sorting

of:

- array with random integers

- array with already sorted integers

Based on you research explain you expectations of the results. Do Excel tables and charts, which represent your results - that means the time taken for a certain case to finish

(you can use a separate library - like time.h) and the sizes of the Stack and the Heap\*.

Compare the results to the expected ones. Do a final conclusion for the memory usage and the sizes of the program segments.

Explain what are the best and the worst case scenarios and why.

Bonus: Do the same task for "Merge Sort" - how did the results change. Why?

\*Note - create you own library, that checks the memory usages and find memory leaks. Create your own design and implement it as best as you can for the given time of one week.

\*\*Attention - upload at least the report file. The source code of the algorithms is not required, but the code of the memory-check library is! Upload whatever else feels necessary.

//

<https://en.wikipedia.org/wiki/Sorting_algorithm>

Quicksort :

<https://upload.wikimedia.org/wikipedia/commons/6/6a/Sorting_quicksort_anim.gif>

<http://www.algolist.net/Algorithms/Sorting/Quicksort>

Mergesort:

<https://upload.wikimedia.org/wikipedia/commons/c/cc/Merge-sort-example-300px.gif>

time.clock();