Programming Languages Comparative Analysis Report

# Used metrics

* Execution time (measured from the program using language’s methods)
* Consumed memory during execution (Heap + Stack)

# Language performance

**JavaScript**

JS takes time to finish longer than other languages in almost all cases. In some algorithms (*counting sort, insertion sort*) it is pretty close to Java and Kotlin, but in all cases JS is behind Swift, C and C++. It absolutely beats Kotlin in Binary Tree insertion, merge sort and insertion sort algorithms.

**Swift**

Swift runs better than other languages except C/C++ in all cases. The same situation with memory - less than others except C/C++. Looks like an ideal language for performance applications and which is much friendlier than C or C++.

**C**

Absolute winner for performance. In all cases other languages are far behind it. Consumes in average 3 times less memory than C++, 11 times less than Swift and 30-80 times less than other languages.

**C++**

High performance language as well. Runs much faster than other languages except for C. Memory consumption is almost the same in all sorting algorithms (around 3,5Mb).

**Kotlin**

Kotlin execution time is pretty close to Java except for some cases when it has the worst result (25ms in mergeSort vs. 3ms for Java, 27ms in BST Insert vs. 2ms for Java). Memory consumption, on the other hand, is always nearly 2 times less than Java.

**Java**

Moderate performance results. In all cases faster than JS except for insertionSort where it has the worst result among others. Compared to Kotlin, it is faster in 50% of cases. But has definitely the biggest memory consumption - nearly 85Mb for each algorithm.

# Code complexity

Since it is subjective for each programmer and depends on one’s primary language, let’s take as a criteria how simple the code can be read. Obviously, JS is the most descriptive language because of its lack of types (code is much cleaner and generic) and short syntax constructions. Next are Swift and Kotlin since it is hard to compare them because of their syntax similarity, although a lot of labels in function arguments in Swift sometimes make it a bit harder to read.

Other languages are really hard to compare. C has a simple syntax but without a lot of standard methods like in other languages it requires more lines of code for the similar tasks, so you have to keep track of the main idea while traversing through little algorithms. C++ is more powerful, but has a lot of language details and sometimes complicated syntax to read, which requires a high skill and knowledge of language to write efficiently. Java just has too many words, and it becomes a problem to see the logic behind them, especially without comments or docs.

# Different input data

All above results were received with an array of 1000 elements.

If we increase it to 10000 elements, all conclusions are almost identical (only increased in magnitude), but some interesting moments can be pointed out. First, Swift lost a lot of performance, with execution time close to JavaScript (in case if insertion sort is even 4 times slower). Second, JS run time explodes in BST insertion (577ms compared with the next slowest Kotlin 27ms). It may signalize some bugs in either measurements or algorithms implementations and need to be studied.