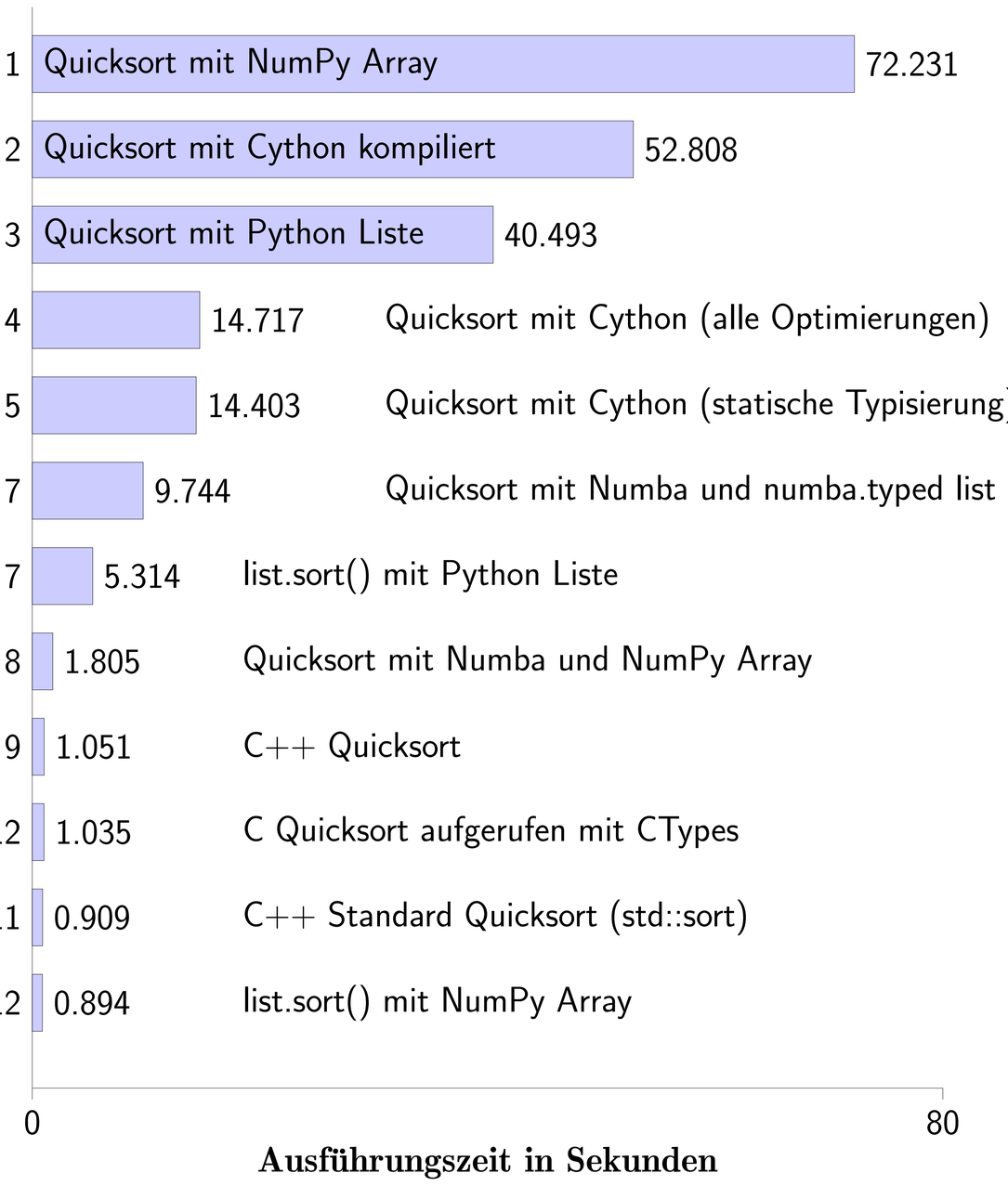
Supersonic Algorithms

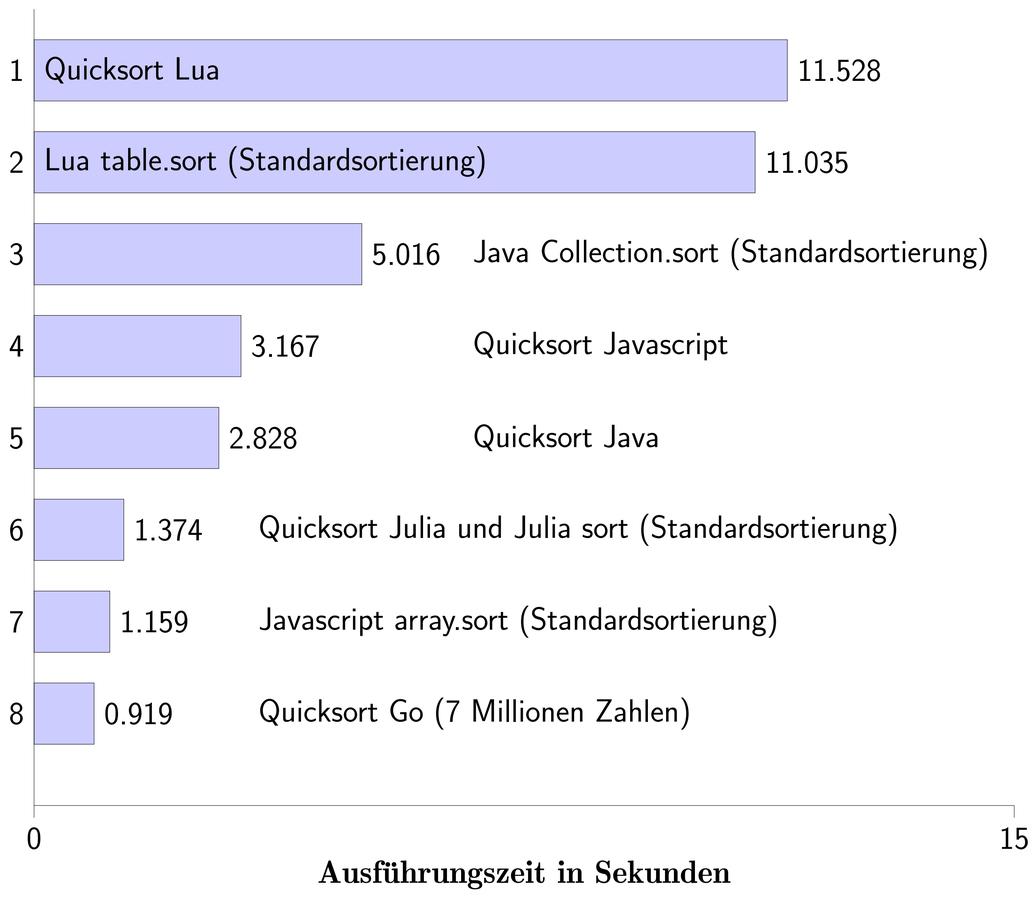
Anton Rodenwald (18), Schillerschule Hannover

Quicksort Implementationen in Python mit verschiedenen Optimierungen und Vergleich mit C++

In diesem Diagramm sind meine verschiedenen Versionen der Quicksort in Python, die standardmäSSig verfügbaren Sortierfunktionen von Python und C++ sowie eine in C++ implementierte Quicksort zu sehen. Alle diese Versionen lieSS ich eine vorher generierte, unsortierte Liste aus 10 Millionen zufälligen Zahlen sortieren. Meine im Informatikunterricht entwickelte Version (3) brauchte ca. 41 Sekunden zur Sortierung. Überraschend war für mich dann, dass Versionen mit Cython und NumPy (1, 2) teils länger brauchten. Dies hatte ich nicht erwartet. Ich ging eigentlich davon aus, dass es nur positive Veränderungen geben würde, also meine Implementationen nur schneller werden würden. Als ich dann Cython zur statischen Typisierung nutzte, zeigte sich ein guter Zuwachs an Performance (5). Weitere Optimierungen mit Cython (4) schienen aber keinen Effekt zu haben, was mich wunderte. Da Cython vor der Ausführung kompiliert wird, hatte ich eigentlich mit einer gröSSeren Geschwindigkeitserhöhung gerechnet, was nicht der Fall war. Ungefähr 4.5x schneller war meine Version mit Numba und der Numba Typed List (6). Nutzte ich Numba nun in Kombination mit einem NumPy Array (8), so erhöhte sich die Geschwindigkeit nochmal um ein Vielfaches. Meine Version mit Numba und NumPy (8) war sogar schneller als die normale Sortierfunktion für Python Listen (7). Dies bestätigte meine Erwartungen, dass NumPy und Numba einen groSSen Boost in Performance bringen, doch es überraschte mich, dass dieser so groSS war. Einzig schneller waren nur einige C/C++ Versionen (9, 10, 11) und die standardmäSSige Sortierfunktion für NumPy Arrays, was ich so erwartet hatte. Meine in C++ implementierte Quicksort (9) war dabei ähnlich schnell wie die standardmäSSige C++ Sortierfunktion (11). Wichtig zu erwähnen zur Version mit CTypes (10) ist, dass die Konvertierung einer Python Liste in ein für C verständliches Format auch nochmal ca. 2 Sekunden Zeit kostete. Es zeigt sich, dass NumPy eine genauso schnelle Sortierung wie C++ ermöglicht. Dies überraschte mich sehr, weil es meiner Hypothese und den Foren-Beiträgen, die ich gelesen hatte, komplett widersprach. Ich fragte mich, wie sich die Versionen 3 und 12 unterschieden. Um mir diese Frage zu beantworten stellte ich einige Überlegungen an, die ich später diskutieren werde.



Quicksort Implementationen in weiteren Sprachen im Vergleich



Neben C++ und Python interessierte mich auch, wie schnell das Sortieren mit einer Implementation der Quicksort in anderen Sprachen möglich ist. Ich nahm dabei aufgrund meines limitierten Wissens in diesen Sprachen keine Optimierungen vor. Dies tat ich auch, weil mich interessierte, wie schnell die nativen Implementationen der Quicksort in diesen ist. Es fällt direkt auf, dass Lua (1, 2) sowohl mit der Quicksort als auch mit der standardmäSSigen Sortierung im Vergleich sehr schlecht abschneidet. Darauf folgt die standardmäSSige Sortierung in Java (3). Diese ist überraschenderweise langsamer als meine Quicksort Implementation in Java (5). Zwischen diesen beiden liegt meine Quicksort in Javascript. Darauf folgt dann Julia (6), wo meine eigene Quicksort fast genauso schnell war wie die standardmäSSige Sortierfunktion. Einzig schneller sind nur die standardmäSSige Sortierfunktion in Javascript (7) und meine Quicksort Implementation in Go (8). Diese ist allerdings nur bedingt vergleichbar, da ich aufgrund von beschriebenen Schwierigkeiten nur 7 Millionen Zahlen sortieren konnte. Meine Go-Variante lässt sich also nicht als schnellste Version bezeichnen.

C++ Radixsort Implementationen

In den vorigen Teilen ging es um die Optimierung der Quicksort. Dieser Abschnitt behandelt die Radixsort, ist also nicht mit den Ergebnissen und Überlegungen von davor vergleichbar. Es wurden hier nicht nur Optimierungen, sondern auch andere Algorithmen genutzt. Der Grund warum ich mich auch mit der Radixsort beschäftigte ist, dass ich den schnellsten Weg zur Sortierung