

Luc Levesque
1238403

	Recursive Quicksort	Iterative Quicksort
Time 1	0.0089	0.0087
Time 2	0.0088	0.0086
Time 3	0.0086	0.0083
Average	~0.00876	~0.00853

The execution times of iterative and recursive quicksort are quite similar. However, after multiple tests, the iterative implementation was faster by around 2.7% on average.

So yes, the iterative method was faster across all 3 test runs on quicktest.txt. The recursive method requires repeated function calls. While modern processors can handle this much more efficiently now, the cumulative function calls clearly add extra time to processing as each call has the overhead of parameter passing, stack frame creation and return address management.

The speed is attributable to modern processors as both algorithms have the same time complexity of $O(n \log n)$, so the base algorithm remains the same. Modern processors have several optimizations that minimize the disadvantages of recursion and the very small difference of speed demonstrates that modern processors handle recursion much more efficiently than they did historically.

I think C would be faster but not by a large amount. C provides more control over memory management which can help with the run time of the quicksort algorithm. However, Fortran's operations are well optimized by compilers like gfortran that even the playing field. So we could potentially see a 1-2% decrease in run time in C.