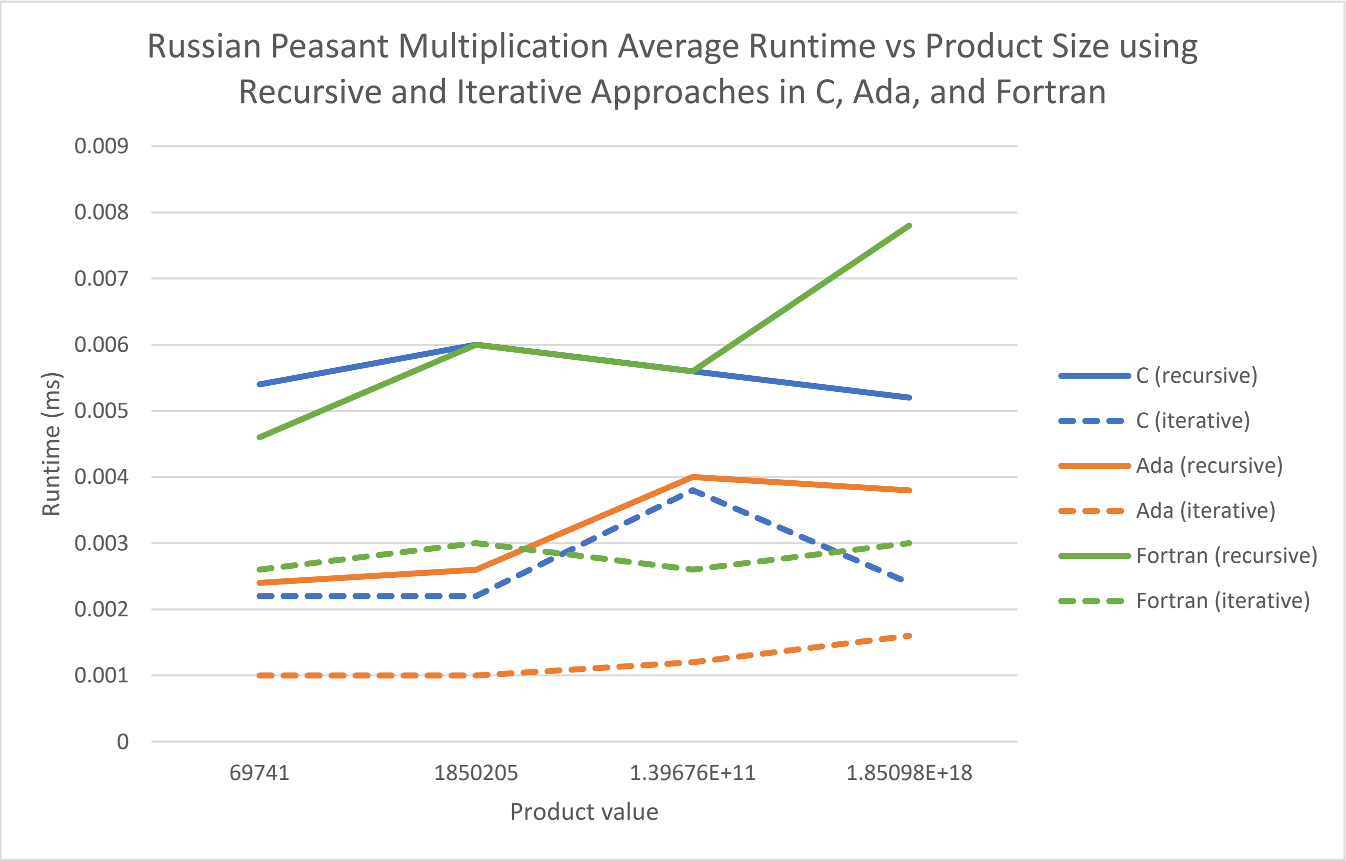
Shawn Kaplan April 6, 2020

Testing – Results and Analysis

In order to test C, Ada, and Fortran’s recursive and iterative approaches to the Russian Peasant Multiplication algorithm, I made a simple script to run each function 5 times per input-set (multiplier and multiplicand), using 4 different input-sets of different sizes, with products ranging from 70,000 to 1.85e18. I then used Excel to organize the data and calculate the average runtime of each function for each product. The results are shown below:

	AVERAGE RUNTIMES					
Product	C (recursive)	C (iterative)	Ada (recursive)	Ada (iterative)	Fortran (recursive)	Fortran (iterative)
69741	0.0054	0.0022	0.0024	0.001	0.0046	0.0026
1850205	0.006	0.0022	0.0026	0.001	0.006	0.003
1.397E+11	0.0056	0.0038	0.004	0.0012	0.0056	0.0026
1.851E+18	0.0052	0.0024	0.0038	0.0016	0.0078	0.003

The table was then graphed to make it easier to see the trends in the data:



Looking at the above graph, it is apparent that the iterative function of each language is significantly faster than its recursive counterpart, with the recursive functions being 2-3x slower in almost every run. We can also see that Ada's iterative function has the fastest runtime out of all the languages, and its recursive function is faster than the other recursive functions. On the slower end, we see that Fortran's recursive algorithm has peak runtime when given the largest inputs, while the other languages' recursive functions see little to no change in their runtimes.

Though the amount of data collected for this mini experiment is limited, it is clear that iteratively performing Russian Peasant Multiplication is much faster than doing it recursively. It is also clear that Ada runs this algorithm at a higher speed than C and Fortran across a wide range of product sizes. This comes as a surprise to me because Fortran is supposed to be a superior language for number processing and mathematical calculations and C is supposed to compile directly into highly efficient machine code. I suspect compiling with different flags may increase the speed of some of these programs, though I am not sure how much of an impact it would have on the overall results.