

Lab 10 Analysis Document –

Hypothesis: I expect that any recursive functions will have time that grows much more rapidly than the iterative. Following this idea, I feel like the double recursion will be the slowest because of the 2 recursive tree calls in the function. I think the tail recursive function will be the next slowest prior to the optimization due to the multiple recursive function calls. Next the single recursion, and the fastest would be the iterative function.

Results:

Prior to Optimization (-O3):

	1	5	25	50	100	200	400	1000	10,000	100,000
Iterative	0	1 e-6	1 e-6	2 e-6	2 e-6	3 e-6	5 e-6	9 e-6	9.1 e-5	8.9 e-4
Single Recursion	1 e-6	1 e-6	1 e-6	1 e-6	2 e-6	2.2 e-5	9 e-6	2.1 e-5	2.79e-4	.003135
Tail Recursion	1 e-6	1 e-6	1 e-6	2 e-6	2 e-6	5 e-6	7 e-7	1.9 e-5	2.12e-4	.0031
Double Recursion	1 e-6	1 e-6	2.65 e -3	256.623	N/A					

Post Optimization:

	1	5	25	50	100	200	400	1000	10,000	100,000
Iterative	1 e-6	1 e-6	1 e-6	1 e-6	1 e-6	1 e-6	1 e-6	3 e-6	2.1 e-5	2.36 e-4
Single Recursion	1 e-6	1 e-6	1 e-6	1 e-6	1 e-6	1 e-6	1 e-6	1 e-6	1 e-6	1 e-6
Tail Recursion	1 e-6	1 e-6	1 e-6	1 e-6	1 e-6	1 e-6	1 e-6	1 e-6	1 e-6	1 e-6
Double Recursion	1 e-6	1 e-6	6.18 e -4	66.0736	>25min	N/A				

It's clear in the un-optimized code that the double recursion grows exponentially, and the trend follows into the optimization. The time cuts significantly, but I was unwilling to run the code any longer after I cooked dinner, ate dinner, cleaned up after dinner and the double recursive calling was still ongoing. This was as I expected it.

The iterative code tracks the same magnitude in both pre and post optimization. In the unoptimized code, the speed was the quickest as I predicted but it did not maintain that title in the optimized code.

The single recursion and tail recursion performed the same in both the pre and post optimized code. It was about what I expected, but I did not expect the optimization to basically rid the code of time. I even tried 1,000,000 for the Fibonacci index to solve, and the code still ran in 1 e-6 amount of time.