

Robert Elsom

3/12/19

Recursive vs Iterative Analysis

Hypothesis:

The purpose of Lab 10 is to test the execution speeds of a recursive vs iterative function. In this test, we are using the Fibonacci series. I believe that the initial test, with a low number of terms will be rather similar in speed. However, as the number of terms being used increases, the speed of the iterative function will be lower, and the more terms, the larger the gap between the two functions. I believe this is because the more terms being tested, the more times the recursive function will have to call itself. For example, if my understanding is correct, 50 terms means the recursive function will call itself 2^n times or 1,125,899,906,842,624 times. However, the iterative function will only have to run n times, which is 50 total times.

Data:

Note: Faster time is highlighted

Number of Terms	Recursive Time (in nanoseconds)	Iterative Time (in nanoseconds)	Difference (in nanoseconds)
1	1244	882	362
10	3038	930	2108
20	191,931	973	190,958
30	23,968,795	1053	23,967,742
40	$1.79596 * 10^9$	1130	$1.79596 * 10^9$
50	$1.93121 * 10^{11}$	1268	$1.93121 * 10^{11}$

Conclusion:

As the data shows, when using low terms, the run times were very similar with a difference of 362 nanoseconds. However, as the number of terms increased, the run times of the recursive time grew exponentially while the iterative time still stayed very small. Therefore, in terms of execution time, iterative functions greatly outperform recursive functions.