

Lab 10 - results, comparison and analysis

Measurement results:

The following chart shows the difference of running time between the recursive and non-recursive (iterative) algorithms:

N	Recursive time	Iterative time
1 – 20	0.001	0.001
21 – 25	0.002	0.001
26	0.003	0.001
27	0.005	0.001
28	0.008	0.001
29	0.011	0.001
30	0.017	0.001
35	0.153	0.001
40	1.618	0.001
45	18.295	0.001

Comparison:

There is a very clear and apparent difference of running time between the recursive and iterative algorithms. As seen in the chart above, the running time among the two algorithms were the same when N was in the range of 1 to 20. As soon as N was 21 or greater, the time difference between the two algorithms started to show. The running time of the iterative algorithm remained constant at ~ 0.001 while the running time grew for the recursive algorithm each time N increased. As soon as N was in the mid to upper 40's, the recursive algorithm took so long to complete that the program would simply sit and run with a blank screen while trying to complete the algorithm. However, when the program finally did finish computing the recursive algorithm, the iterative algorithm would still take only ~ 0.001 seconds to complete. This is a tremendous difference in running time between these two algorithms. In fact, I had to comment out the recursive algorithm and solely just test the iterative algorithm to see at what N would the running time start to change for the iterative algorithm. Roughly around when N = 200,000 is when the running time of the recursive algorithm started changing from 0.001 to 0.002. There is no question about it that the iterative algorithm has a much quicker run time than the recursive algorithm.

Analysis:

As pointed out in my comparison between the recursive and iterative algorithms, the iterative algorithm is MUCH faster than the recursive algorithm. The reason for this is because with iterative algorithms, you simply execute a new repetition of the code that is in the body of a loop. However, with recursive algorithms, each repetition is a function call. This means that each function call pushes the existing program state onto the call stack. This is a memory operation and memory speeds are MUCH slower than processor speeds. This is what causes the big difference in running time between these two algorithms.