CS 201, Spring 2017

Homework Assignment 2

In this assignment, we evaluated the complexity of the calculating "Fibonacci Numbers" with C++. There are many methods for calculating it however we tested 2 ways; one of them is recursive way which comes from its definition F(n) = F(n-1) + F(n-2), other way is iterative solution which we add values to the sum. Following table shows the running times according the inputs.

Algorithm 1: An iterative algorithm which works in O(N) time.

Algorithm 2: A recursive algorithm which works in $O(2^N)$ time.

Figure 1: Running times of some inputs

Input Size / Running Time	O(N)	$O(2^N)$
N = 10	0.0004	0.0044
N = 25	0.0004	75
N = 50	0.0008	562170
N = 100000	0.3549	N/A

When input size gets bigger and bigger, algorithm 1 which have a complexity O(N) grows linearly on the other hand algorithm 2 which have complexity $O(2^N)$ grows much faster even the time needed to calculate the 50th fibonacci number is really large comparing to algorithm 1 which uses the iterative way.

Figure 3: Running time graph for the recursive solution

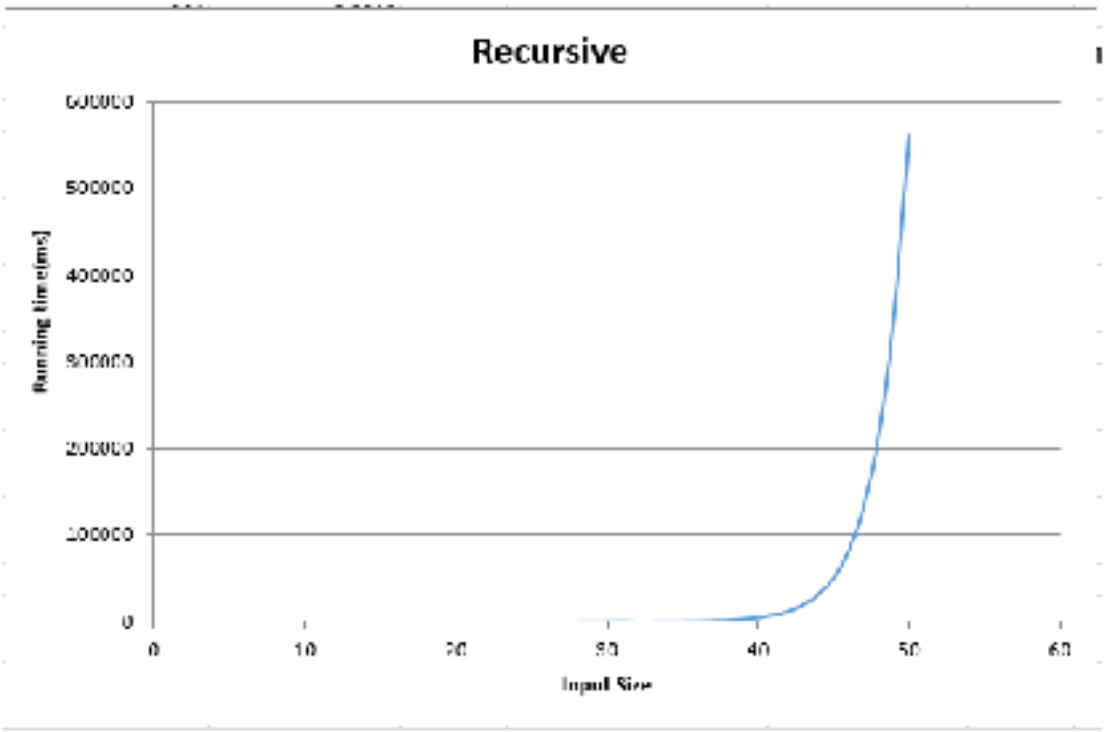
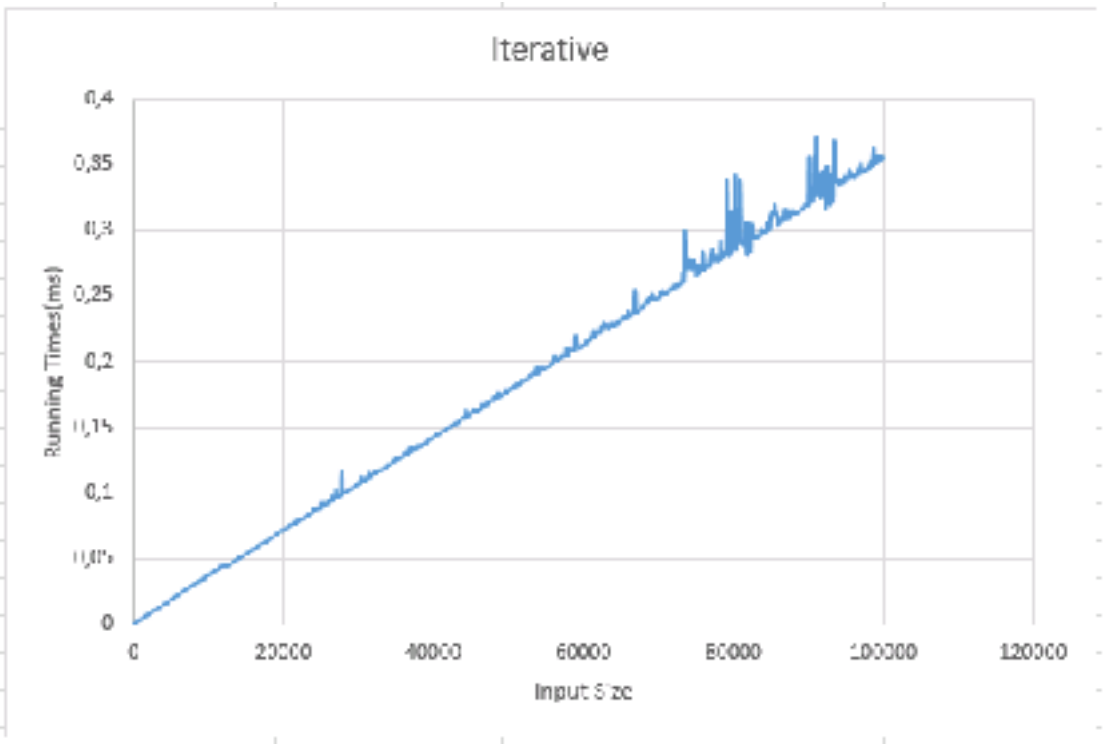


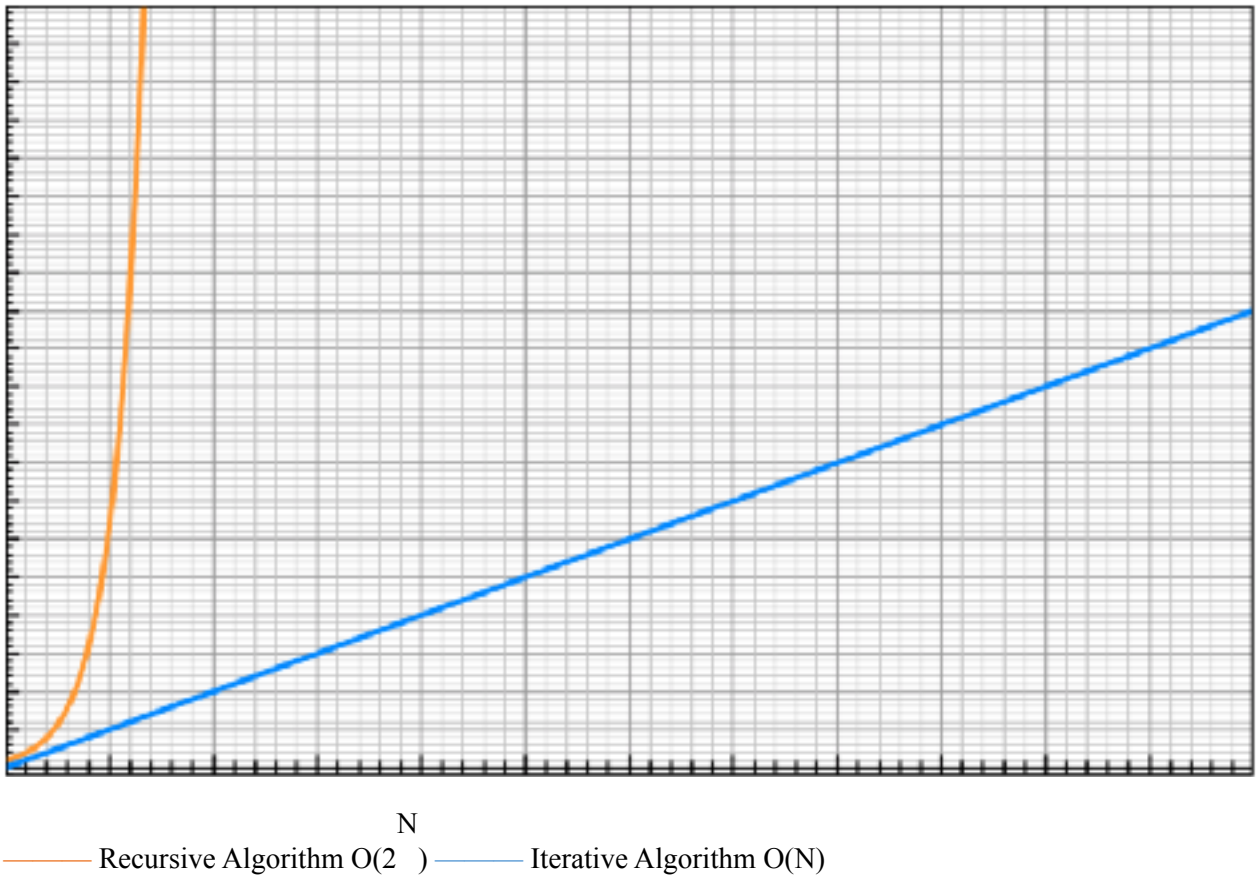
Figure 4: Running time graph for the iterative solution



These graphs are obtained by my computer. Recursive algorithm could calculate the 50th fibonacci in an appropriate time, incrementing input size really slows the operation time. However, calculating the 100.000th fibonacci number is really easy as it could be seen from the graph. It took only 0.35 milliseconds while recursive solution gives 50th fibonacci in nearly 500.000 milliseconds. Also when we compare these graphs with theoretical ones, it can be

seen that results are similar.

Figure 3: Space complexity graph for these algorithms



As a consequence, according to both theoretical and experimental results, algorithm 1 which uses the iterative way to calculate fibonacci numbers has time complexity O(N) and algorithm 2 which uses the recursive way has time complexity $O(2^N)$. Therefore iterative algorithm is much faster than the recursive algorithm because its linear while recursive algorithm is exponential. It can be inferred from the experiments of these algorithms and resulting graphs in the above.

Computer Specifications

MSI GS-70 2QE Laptop

- Intel Core i7 4720HQ Quad Core CPU - 2.6GHz - Turbo up to 3.6GHz
- 1 x 1TB HDD (7200RPM) + 2 x 128GB SSD
- 2 x 8GB DDR3
- 17.3" LCD LED Full HD Screen (1920x1080)
- NVIDIA GTX-970M 3GB Graphics Card

Kerem Ayöz

21501569 Section:1