Name : Hassan Raza

ID : 21701811

CS201-3

Analyzing the Growth Rate of two different algorithms used for finding the Fibonacci numbers.

Computer Specifications:

Processor: Intel® Core™ i7-7500U CPU@ 2.7GHz 64 bit

RAM: 8.00GB

OS: Windows 10 pro

Table:

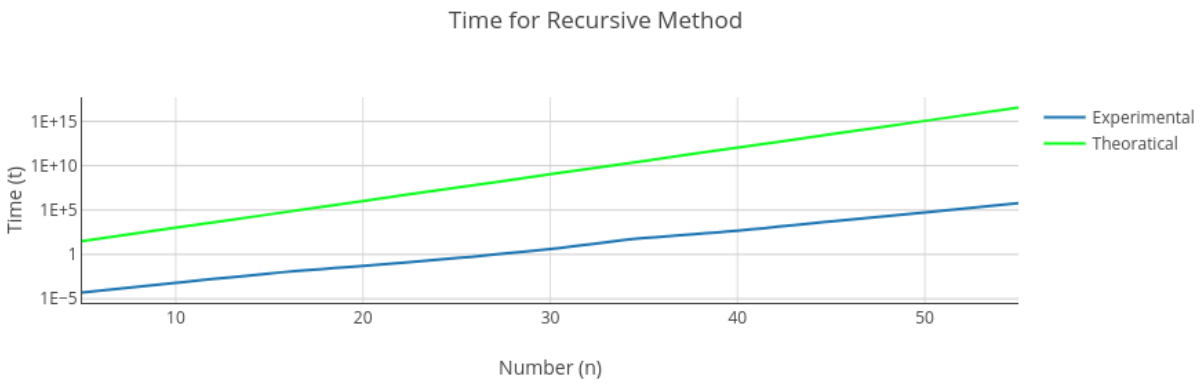
Shows the time taken for both recursive and iterative methods for different values of n, which is number of Fibonacci numbers calculating.

Number n	Recursive time $O(2^n)$ (ms)	Iterative time O(n) (ms)
5	.00005	.000009
10	.0006	.00002
15	.0073	.00003
20	.0503	.00005
25	.3955	.00006
30	4	.00007
35	69	.00009
40	458	.00010
45	5096	.00011
50	53717	.00012
55	618794	.00013
100		.0003
1000		.008
10000		.072
100000		.522
1000000		3
10000000		30
100000000		290
250000000		772
500000000		1488
750000000		2215
1000000000		3004

Plots:

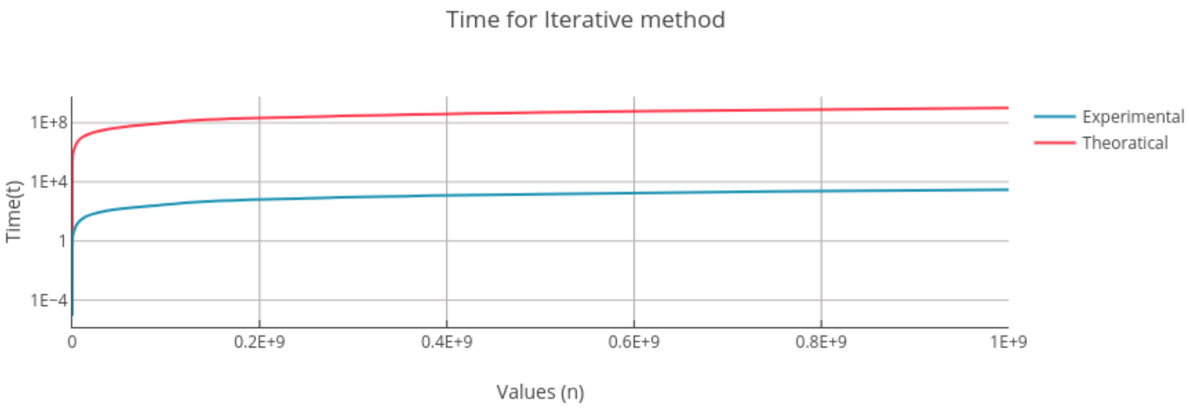
Plot showing the time taken for the recursive method. Here log of time is along y-axis:

Plot 1:



Plot showing the time taken for iterative method. Here log of time is along y-axis:

Plot 2:



Discussion:

For the plot 1 which is for the recursive method we can see that both expected growth rate of time and actual growth are similar. For this method growth rate is exponential i-e 2^n which is visible from the graph as well, which is straight line. Similarly for the plot 2 which is for iterative method we have a similar kind of observation that both expected and actual growth of time is same. For this method growth rate is linear which is simply n and indicating this in graph the line gradually becomes almost parallel to x-axis.

We can see that the growth rate of recursive method is much greater than the iterative method. The reason for this is that in iterative methods we have a linear relation with time and for every extra call for loop we will have to go through loop just one more time whereas in recursive methods relation with time is exponential as for each extra call we will go through method n more times therefore the time for this is greater than iterative ones.