Name: Omar Ahmed UCID: 30154382

PA-1 – Comparing Algorithms – Report

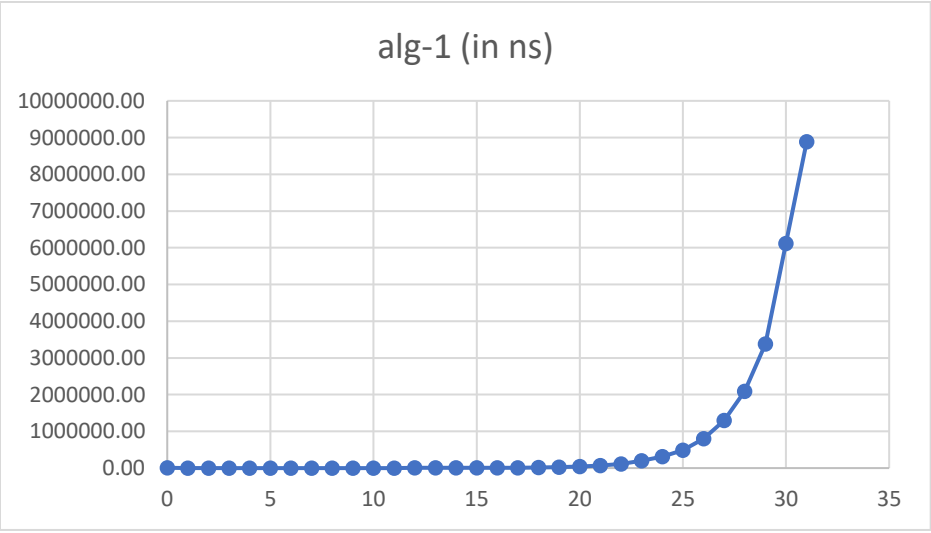
Question 4:

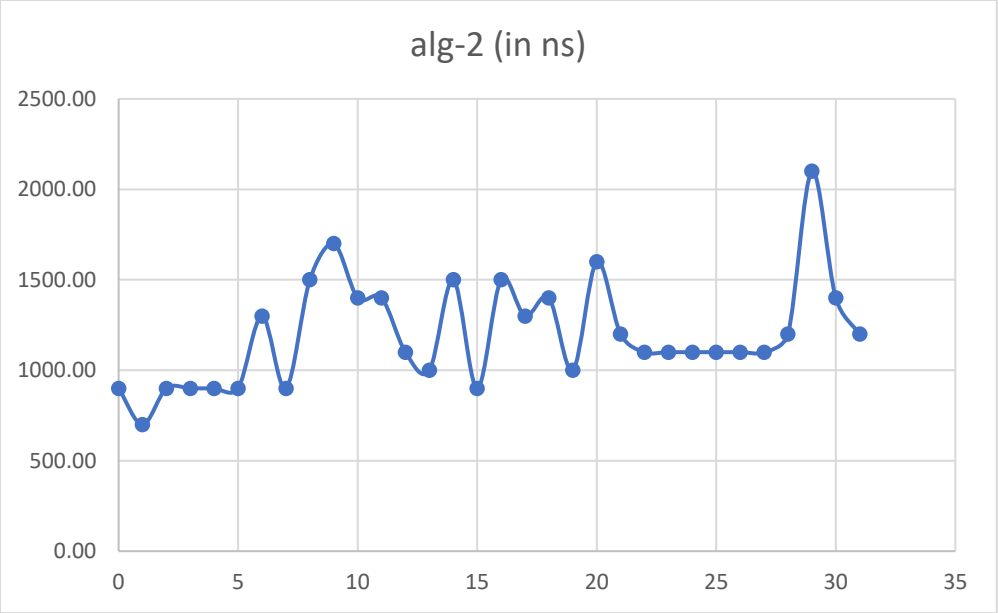
Algorithm 1 (Recursive): The recursive algorithm exhibits an exponential growth in execution time as `n` increases. This algorithm might be used for educational purposes or when `n` is guaranteed to be very small. It's simple and demonstrates the concept of recursion, but due to its inefficiency, it is not practical for large values of `n`.

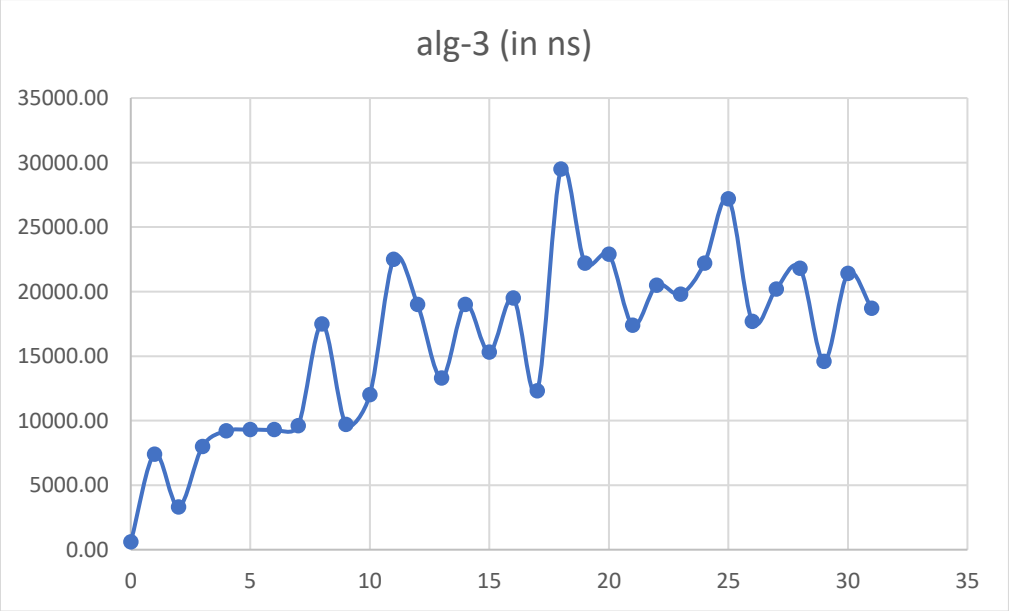
Algorithm 2 (Iterative): The iterative algorithm shows linear growth in execution time, which makes it very practical for small to moderately large values of `n`. It has the advantage of being simple to understand and implement, and it requires constant memory space. This algorithm would be appropriate for most applications where the Fibonacci number computation is not a performance bottleneck and when `n` is not so large that the computation time becomes too long.

Algorithm 3 (Matrix Exponentiation): The matrix exponentiation algorithm, despite some fluctuations, generally shows a slow growth rate in execution time, indicative of logarithmic complexity. This algorithm is the most efficient of the three for large values of `n`. It's particularly useful in performance-critical applications or where `n` is large, and you need to compute Fibonacci numbers rapidly. However, it is more complex than the iterative method and might be overkill for small values of `n` or when the computation of Fibonacci numbers is not the performance bottleneck.

Plots:




alg-2 (in ns)
2500.00
2000.00
1500.00
1000.00
500.00
0.00
0
5
10
15
20
25
30
35


alg-3 (in ns)
35000.00
30000.00
25000.00
20000.00
15000.00
10000.00
5000.00
0.00
0
5
10
15
20
25
30
35