Aravind Krishna
20110021
COA Assignment 1

Qn1:

First 50 Fibonacci numbers using recursion:
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946 17711 28657 46368 75025 121393 196418 317811 514229 832040 1346269 2178309 3524578 5702887 9227465 14930352 24157817 39088169 63245986 102334155 165580141 267914296 433494437 701408733 1134903170 1836311903 2971215073 4807526976 7778742049
CPU time: 84.6384 seconds

First 50 Fibonacci numbers using loop:
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946 17711 28657 46368 75025 121393 196418 317811 514229 832040 1346269 2178309 3524578 5702887 9227465 14930352 24157817 39088169 63245986 102334155 165580141 267914296 433494437 701408733 1134903170 1836311903 2971215073 4807526976 7778742049
CPU time: 8.8e-06 seconds

First 50 Fibonacci numbers using recursion with memoization:
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946 17711 28657 46368 75025 121393 196418 317811 514229 832040 1346269 2178309 3524578 5702887 9227465 14930352 24157817 39088169 63245986 102334155 165580141 267914296 433494437 701408733 1134903170 1836311903 2971215073 4807526976 7778742049
CPU time: 4.56e-05 seconds

First 50 Fibonacci numbers using loop with memoization:
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946 17711 28657 46368 75025 121393 196418 317811 514229 832040 1346269 2178309 3524578 5702887 9227465 14930352 24157817 39088169 63245986 102334155 165580141 267914296 433494437 701408733 1134903170 1836311903 2971215073 4807526976 7778742049
CPU time: 1.67e-05 seconds

1. Recursive Method (Baseline)
   - CPU Time: 84.6384 seconds

2. Loop Method
   - CPU Time: 8.8e-06 seconds
   - **Speedup: ≈ 9,618,000**

3. Recursive Method with Memoization
   - CPU Time: 4.56e-05 seconds
   - **Speedup: ≈ 1,856,535**

4. Loop Method with Memoization
   - CPU Time: 1.67e-05 seconds
   - **Speedup: ≈ 5,067,137**

Aravind Krishna
20110021
COA Assignment 1

Qn2:

Bucket 1 - C++

Matrix multiplication for size 64x64 (Integer):
Elapsed time: 0.00124548 seconds

real 0m0.011s
user 0m0.000s
sys 0m0.005s

Matrix multiplication for size 64x64 (Double):
Elapsed time: 0.0013017 seconds

real 0m0.004s
user 0m0.003s
sys 0m0.000s

Matrix multiplication for size 128x128 (Integer):
Elapsed time: 0.0129012 seconds

real 0m0.014s
user 0m0.014s
sys 0m0.000s

Matrix multiplication for size 128x128 (Double):
Elapsed time: 0.0112976 seconds

real 0m0.013s
user 0m0.013s
sys 0m0.000s

Matrix multiplication for size 256x256 (Integer):
Elapsed time: 0.0916474 seconds

real 0m0.094s
user 0m0.085s
sys 0m0.000s

Matrix multiplication for size 256x256 (Double):
Elapsed time: 0.100147 seconds

Aravind Krishna
20110021

real 0m0.102s
user 0m0.102s
sys 0m0.000s
Matrix multiplication for size 512x512 (Integer):
Elapsed time: 0.775512 seconds

real 0m0.787s
user 0m0.780s
sys 0m0.000s

Matrix multiplication for size 512x512 (Double):
Elapsed time: 0.817398 seconds

real 0m0.824s
user 0m0.824s
sys 0m0.000s

Matrix multiplication for size 1024x1024 (Integer):
Elapsed time: 7.02308 seconds

real 0m7.034s
user 0m7.014s
sys 0m0.010s

Matrix multiplication for size 1024x1024 (Double):
Elapsed time: 12.1445 seconds

real 0m12.162s
user 0m12.152s
sys 0m0.010s

Aravind Krishna
20110021
COA Assignment 1

Bucket2 - Python

Matrix multiplication for size 64x64 (Integer):
Elapsed time: 0.06 seconds

real 0m0.397s
user 0m0.545s
sys 0m1.531s

Matrix multiplication for size 64x64 (Double):
Elapsed time: 0.07 seconds

real 0m0.203s
user 0m0.525s
sys 0m1.512s

Matrix multiplication for size 128x128 (Integer):
Elapsed time: 0.48 seconds

real 0m0.615s
user 0m0.961s
sys 0m1.502s

Matrix multiplication for size 128x128 (Double):
Elapsed time: 0.55 seconds

real 0m0.684s
user 0m1.136s
sys 0m1.412s

Matrix multiplication for size 256x256 (Integer):
Elapsed time: 3.93 seconds

real 0m4.069s
user 0m4.564s
sys 0m1.371s

Matrix multiplication for size 256x256 (Double):
Elapsed time: 4.14 seconds

real 0m4.236s
user 0m4.620s
sys 0m1.466s

Aravind Krishna
20110021
COA Assignment 1

Matrix multiplication for size 512x512 (Integer):
Elapsed time: 33.79 seconds

real 0m33.914s
user 0m34.398s
sys 0m1.351s

Matrix multiplication for size 512x512 (Double):
Elapsed time: 36.24 seconds

real 0m36.384s
user 0m36.832s
sys 0m1.416s

Matrix multiplication for size 1024x1024 (Integer):
Elapsed time: 300.83 seconds

real 5m0.985s
user 5m1.345s
sys 0m1.490s

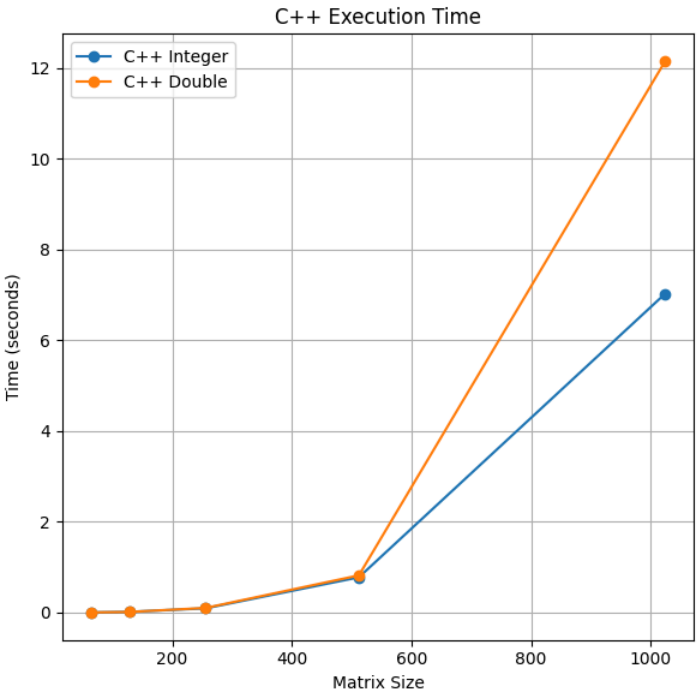
Matrix multiplication for size 1024x1024 (Double):
Elapsed time: 319.81 seconds

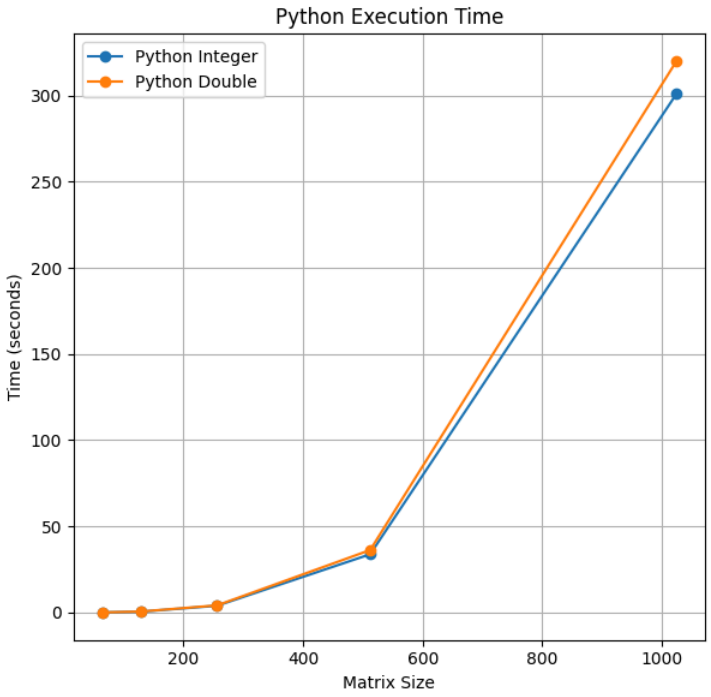
real 5m19.974s
user 5m20.454s
sys 0m1.381s

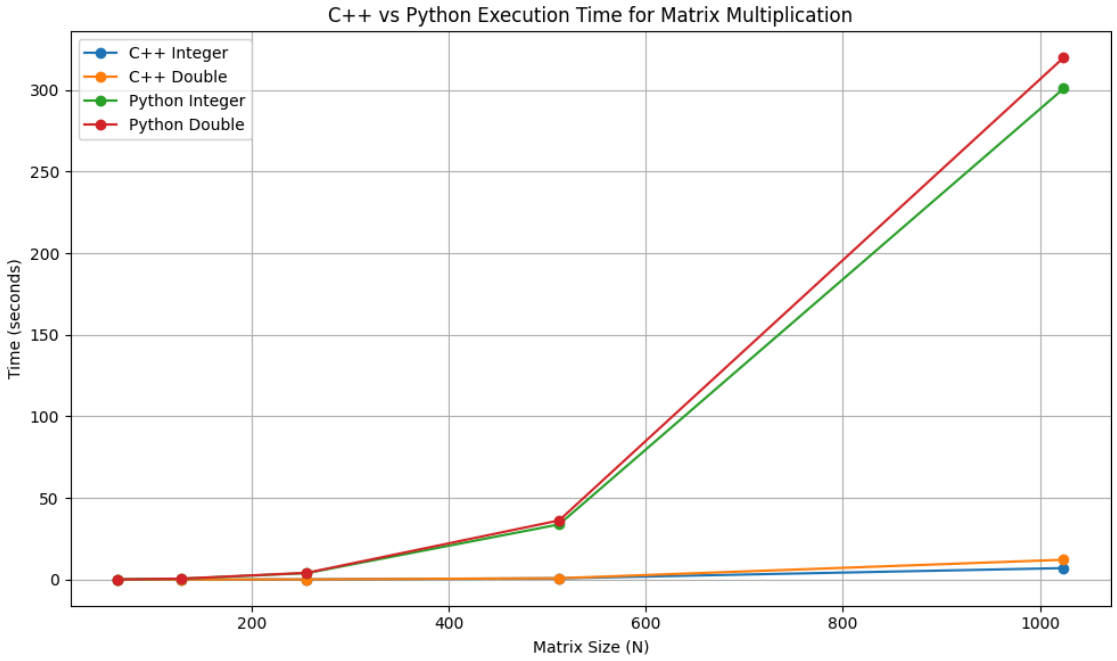
Aravind Krishna
20110021
COA Assignment 1

Comparison:

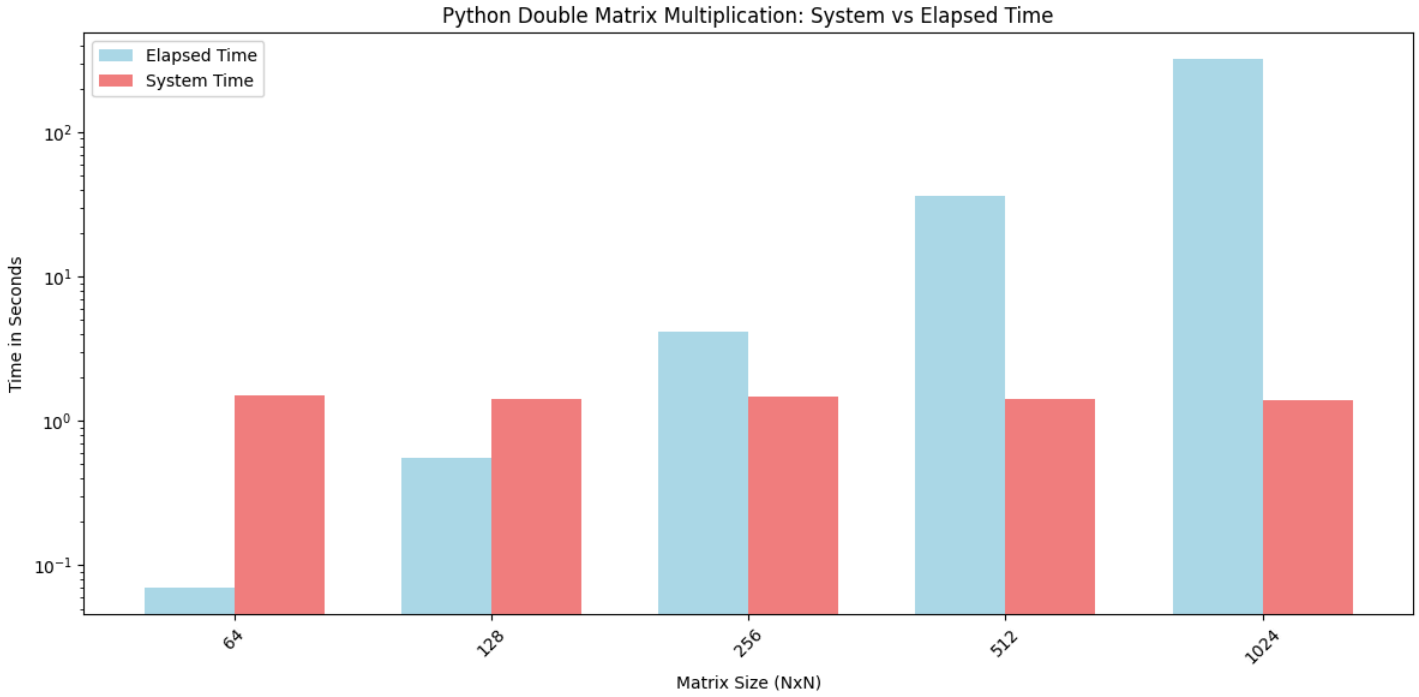
Elapsed time : It is the time taken for multiplication

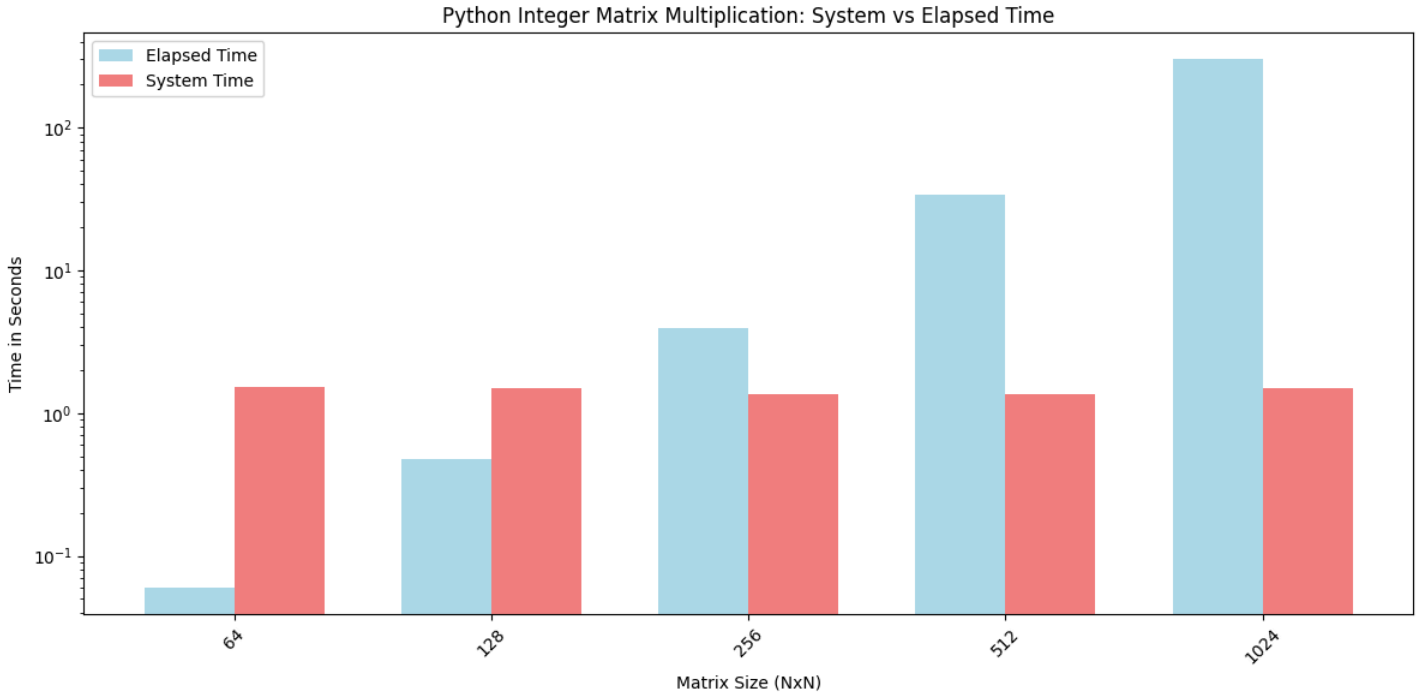




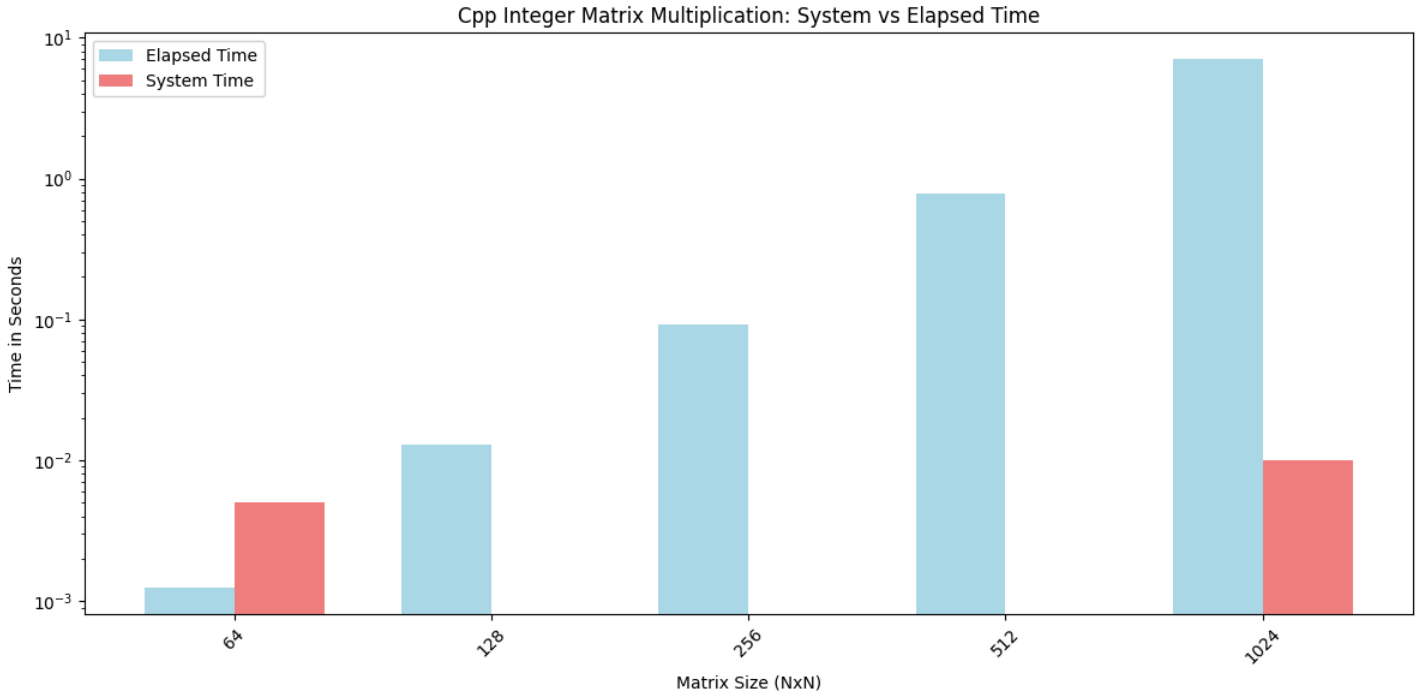


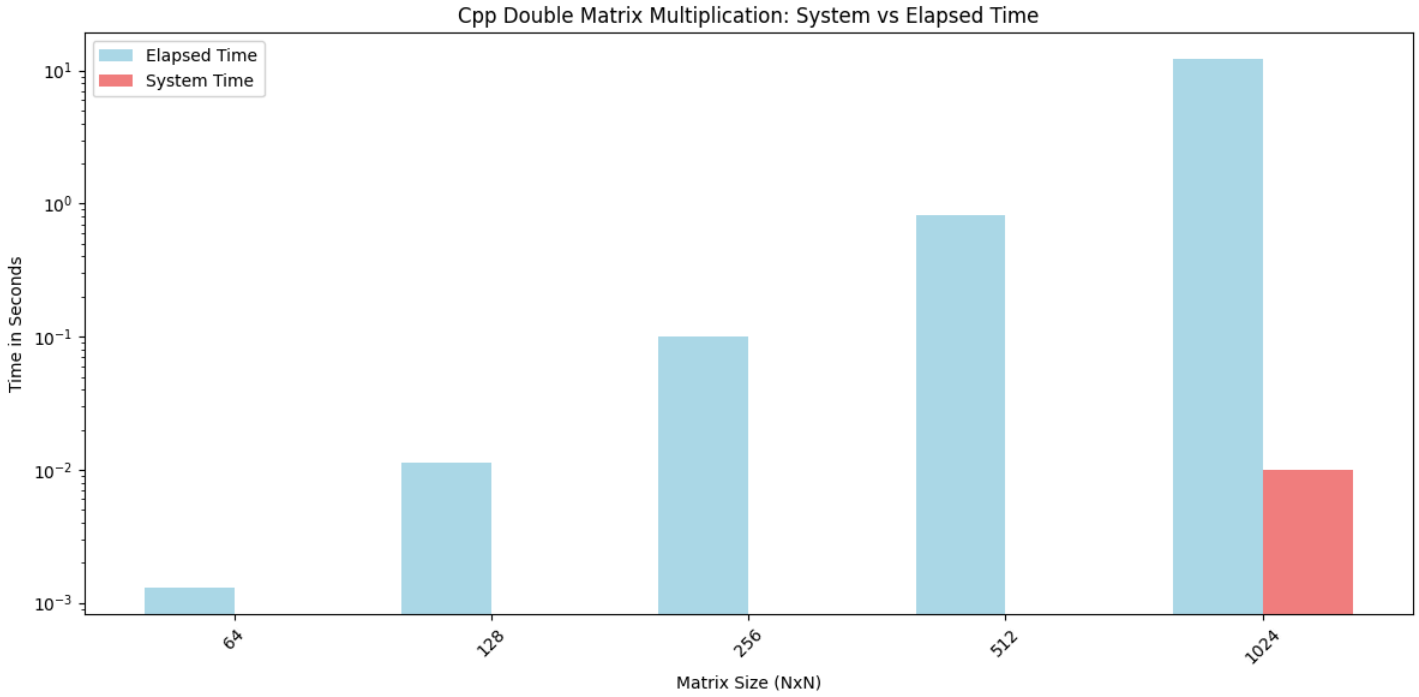
Aravind Krishna
20110021
COA Assignment 1





Aravind Krishna
20110021
COA Assignment 1





Aravind Krishna
20110021
COA Assignment 1

- **The `elapsed` time in C++ is primarily influenced by the computations and less by the system overhead, as C++ is a compiled language with generally lower-level operations.**
- **`System` time is minor relative to `elapsed` time unless the I/O operations or system calls heavily predominate.**

- **In Python, a higher `sys` relative to `elapsed` time is observed due to Python's nature of high-level operations and potential reliance on system resources for handling data and memory management.**
- **Python shows more significant times overall compared to C++ due to its interpreted nature.**