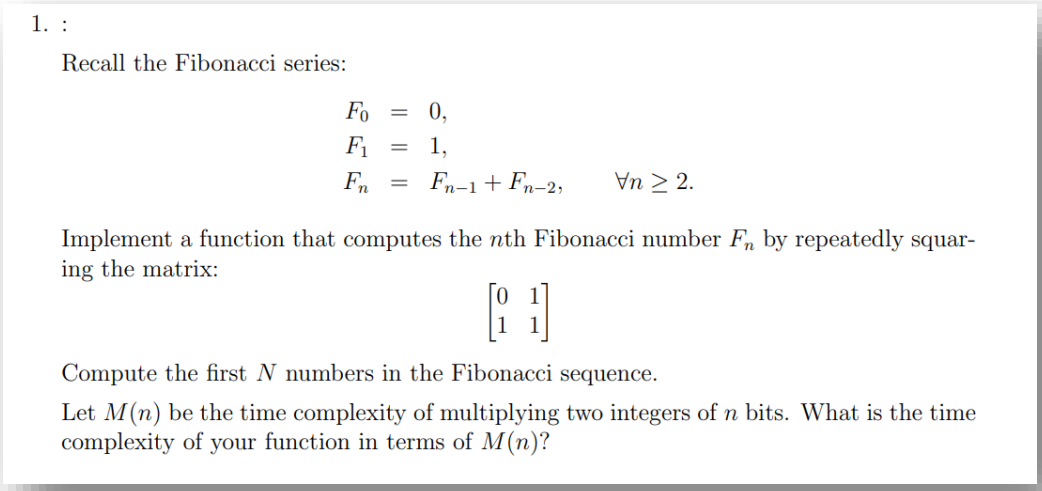
CS222 - Algorithm Design

Dr. Arpita Korwar

Assignment – 4 ***:*** *Fibonacci numbers using repeated squaring of a matrix*

Authors :

- Prakhar Mathur 1906328
- Sanjay Marreddi 1904119
- Rishabh Tripathi 1904129



1. :

Recall the Fibonacci series:

$$\begin{aligned} F_0 &= 0, \\ F_1 &= 1, \\ F_n &= F_{n-1} + F_{n-2}, \qquad \forall n \geq 2. \end{aligned}$$

Implement a function that computes the nth Fibonacci number F_n by repeatedly squaring the matrix:

$$\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$$

Compute the first N numbers in the Fibonacci sequence.

Let $M(n)$ be the time complexity of multiplying two integers of n bits. What is the time complexity of your function in terms of $M(n)$?

- The first 40 numbers Fibonacci sequence are :

  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

- Time complexity of multiplication of two integers of N bits is M(N).
- **n** : represents the n in ***nth Fibonacci Number***.

- The Time Complexity of our function that computes the nth Fibonacci number F_n by repeatedly squaring matrix in terms of $M(N)$ is $\boldsymbol{O(\ M(N)\ \times \log\ (n)\ )}$