Proof of Fibonacci Computation Using Matrix Exponentiation

Mohhamad hayEri

10/14/24

Introduction

The Fibonacci sequence can be computed using matrix exponentiation, achieving a time complexity of $O(\log n)$. We will prove this by following these steps:
1. Matrix representation of Fibonacci numbers. 2. Matrix multiplication. 3. Matrix exponentiation. 4. Final result.

Step 1: Matrix Representation of Fibonacci Numbers

The Fibonacci numbers can be expressed in terms of matrix multiplication as follows:

$$\begin{bmatrix} F(n) \\ F(n-1) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^{n-1} \begin{bmatrix} F(1) \\ F(0) \end{bmatrix}$$

Where:

$$F(0) = 0, \quad F(1) = 1$$

The transformation matrix M is defined as:

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

Step 2: Matrix Multiplication

To multiply two 2×2 matrices A and B:

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \quad B = \begin{bmatrix} e & f \\ g & h \end{bmatrix}$$

The product $C = A \cdot B$ is given by:

$$C = \begin{bmatrix} ae + bg & af + bh \\ ce + dg & cf + dh \end{bmatrix}$$

This multiplication takes constant time, $O(1)$, since it involves a fixed number of operations.

Step 3: Matrix Exponentiation

To compute M^k efficiently, we use the method of exponentiation by squaring:

- If $k = 0$, $M^0 = I$ (the identity matrix).
- If $k = 1$, $M^1 = M$.
- If k is even, $M^k = M^{k/2} \cdot M^{k/2}$.
- If k is odd, $M^k = M \cdot M^{k-1}$.

This method reduces the number of multiplications needed to compute M^k to $O(\log k)$. Each multiplication of two 2×2 matrices takes $O(1)$, so the overall time complexity for matrix exponentiation is $O(\log n)$.

Step 4: Final Result

To find $F(n)$:

1. Compute $M^{(n-1)}$ using the matrix exponentiation method. 2. The top left element of the resulting matrix $M^{(n-1)}$ will be $F(n)$.

Thus, we have shown that the n-th Fibonacci number can be computed in $O(\log n)$ time using matrix exponentiation.

Conclusion

The proof is complete, and we have established that the Fibonacci sequence can be computed efficiently using matrix exponentiation, achieving a time complexity of $O(\log n)$.