Fibonacci Question

Matish Singh Tanwar
AI20MTECH11005

***Abstract*—This document finds out the fibonacci number in O(logn) time.**

Download all latex codes from

https://github.com/Matish007/Matrix−Theory−EE5609−/tree/master/Fibonacci

1 Problem

Find the fibonacci number using Matrix in O(logn) time.

2 Explanation

First thing which come in our mind is the equation of fibonacci number i.e.

$$\boxed{f_n = f_{n-1} + f_{n-2}}. \tag{1}$$

Now,how to write this whole equation in matrix form.Let's try

$$\begin{pmatrix} f_n \end{pmatrix} = \begin{pmatrix} 1 & 1 \end{pmatrix} \begin{pmatrix} f_{n-1} \\ f_{n-2} \end{pmatrix} \tag{2}$$

If we solve the equation (2) we will get the equation (1).
For finding f_n we require two values f_{n-1} and f_{n-2} again we have to calculate for them if further values are not known to us then again calculate them.So,equation (2) is just the matrix form of equation(1).But we cannot calculate the n^{th} fibonacci number in less than O(n) time using equation (2) because we cannot self multiply the matrix $\begin{pmatrix} 1 & 1 \end{pmatrix}$ due to its order.We will modify our equation (2) for finding number in less computation time.But some important things can be inferred out from equation (2) which can be used for determining a matrix which can find n^{th} fibonacci number in less time which are as follows:-

- We need two values f_{n-1},f_{n-2} for calculation of f_n
- We will first try for 2X2 matrix as then self multiplication would be possible.
- If we see equation (2) then first row of matrix $\begin{pmatrix} 1 & 1 \end{pmatrix}$ and first column of matrix $\begin{pmatrix} f_{n-1} & f_{n-2} \end{pmatrix}^T$ is giving us f_n.
- So,our resultant 2X2 matrix should have non zero first column and first row.
- Now L.H.S in equation(2) should be of order 2X1. So,we should add one more entry at L.H.S of equation (2).

- L.H.S contain f_n so we should create a entry of f_{n-1}.
- Now,we should change our equation(2) according to above points
- How our equation is now?

$$\begin{pmatrix} f_n \\ f_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & a_{11} \end{pmatrix} \begin{pmatrix} f_{n-1} \\ f_{n-2} \end{pmatrix} \tag{3}$$

- Comparing R.H.S and L.H.S in equation(3) we get,a_{11}=0
- So our matrix becomes

$$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \tag{4}$$

- Replacing (4) in (3) we get

$$\boxed{\begin{pmatrix} f_n \\ f_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} f_{n-1} \\ f_{n-2} \end{pmatrix}} \tag{5}$$

Since equation (4) matrix is of order 2X2,we can perform multiplication of itself contradictory to $\begin{pmatrix} 1 & 1 \end{pmatrix}$ matrix whose order was 1X2.We cannot multiply that matriix to itself that's why we have to come through this whole process.
If initial values are f_0 and f_1,We can rewrite equation 5 as:-

$$\boxed{\begin{pmatrix} f_n \\ f_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{n-1} \begin{pmatrix} f_1 \\ f_0 \end{pmatrix}} \tag{6}$$

Now,you are thinking about we still have to do matrix multiplication n times,So it is taking O(n).
Well we can calculate $\mathbf{A^n}$ in O(logn) time.Here,$\mathbf{A}$ is a matrix. We can do this by using divide and conquer technique like this:-

- Let n=2^m
- Now divide n by two
- Combine them by using recurrence.
- Now conquer them by multiplying to itself two times

Let's understand by a example:- Suppose n=8,2^m=8
$\implies m = 3$
and we have to calculate $\mathbf{A^8}$
8/2=4
4/2=2
2/2=1=2^0
$\mathbf{A^2}$=$\mathbf{A}.\mathbf{A}$
$\mathbf{A^4}$=$\mathbf{A^2}.\mathbf{A^2}$
$\mathbf{A^8}$=$\mathbf{A^4}.\mathbf{A^4}$
For $\mathbf{A^8}$ literally we required 3 multiplications only and we got $\mathbf{A^8}$. So,we just required O(logn) time to calculate $\mathbf{A^n}$.

Hence,we can find n^{th} fibonacci number in just O(logn) time.