FIT 2004

NOT EXAMINABLE

ARUN KONAGURTHU

NOTES ON Q7 OF WEEK2 PRAC

A Fibonacci Sequence can be described using a second-order recurrence relationship:

$$f_{n+1} = f_n + f_{n-1}$$

A really cute transformation of this 2° relationship to a first-order relationship is using the linear system

$$\underbrace{\begin{bmatrix} f_{n+1} \\ f_n \end{bmatrix}}_{U_n} = \underbrace{\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} f_n \\ f_{n-1} \end{bmatrix}}_{U_{n-1}}$$

From this:

$U_2 = A U_1$, where $U_1 = \begin{bmatrix} f_2 \\ f_1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

$\Rightarrow U_3 = A(U_2) = A^2 U_1$

$U_4 = A(U_3) = A^3 U_1$

$\vdots$

$U_n = A U_{n-1} = A^{n-1} U_1$

Matrix Exponentiation can be decomposed using, what's in Linear Algebra, is called the "EIGENDECOMPOSITION OF A MATRIX"! In other words any square matrix 'A' can be decomposed as:

$$A = S \Lambda S^{-1}$$

where S is the matrix of eigenvectors

S^{-1} is Inverse of S

(Lambda) Λ is the diagonal matrix of Eigen values.

For the Fibonacci sequence, we saw that

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

The Diagonal matrix ' Λ ' of Eigen values can be computed by Solving:

$$|A - \lambda I| = 0$$

$$= \left| \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} - \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix} \right| = \lambda^2 - \lambda - 1 = 0$$

giving the roots

$$\lambda = \frac{1 \mp \sqrt{5}}{2}$$

$$\therefore \quad \Lambda = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} = \begin{bmatrix} \frac{1-\sqrt{5}}{2} & 0 \\ 0 & \frac{1+\sqrt{5}}{2} \end{bmatrix}$$

Eigen vector matrix 'S' can be computed as:

$$(A - \lambda I)\vec{s} = 0$$

$$= \begin{pmatrix} 1-\lambda & 1 \\ 1 & -\lambda \end{pmatrix} \begin{pmatrix} \vec{s}_x \\ \vec{s}_y \end{pmatrix} = 0$$

which gives the relationship

$$\vec{s}_x = \lambda \vec{s}_y$$

Setting $\vec{s}_y = 1$ gives $\vec{s}_x = \lambda$.

So the Eigen vector (unnormalized) is of the form: $\begin{pmatrix} \lambda \\ 1 \end{pmatrix}$.

Normalizing gives: $\begin{pmatrix} \frac{\lambda}{\sqrt{\lambda^2+1}} \\ \frac{1}{\sqrt{\lambda^2+1}} \end{pmatrix}$

Therefore:

$$S = \begin{pmatrix} \frac{\lambda_1}{\sqrt{\lambda_1^2+1}} & \frac{\lambda_2}{\sqrt{\lambda_2^2+1}} \\ \frac{1}{\sqrt{\lambda_1^2+1}} & \frac{1}{\sqrt{\lambda_2^2+1}} \end{pmatrix}$$

So computing any f_N in the Fibonacci sequence can be transformed into computing any

$$U_N = A\, U_{N-1} = A^{N-1} U_1 = S \Lambda^{N-1} S^{-1} U_1$$

Note $U_N = \begin{pmatrix} f_{N+1} \\ f_N \end{pmatrix}$ & $U_1 = \begin{pmatrix} f_2 = 1 \\ f_1 = 1 \end{pmatrix}$

Also note the evaluation of U_N involves matrix multiplication of S with Λ^{N-1}, followed by multiplication with S^{-1}, followed finally by the multiplication of the result with U_1. ~~[illegible]~~

S, Λ, S^{-1} are 2×2 matrises. U_1 is a 2×1 matrix

The only term that depends on 'N' is $\boxed{\Lambda^{N-1}}$.

Since $\Lambda = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \Rightarrow \Lambda^{N-1} = \begin{pmatrix} \lambda_1^{N-1} & 0 \\ & \lambda_2^{N-1} \end{pmatrix}$

λ_1 & λ_2 are real numbers, & the best exponentiation algorithm for a real numbers ~~is~~ of the for x^N is $O(\log N)$

[See: Knuth's Art of Computer programming
Vol. 2: Pages 461–475
(Seminumerical algorithms)]