Algorithm Engineering

Jens K. Mueller

jkm@informatik.uni-jena.de

Department of Mathematics and Computer Science
Friedrich Schiller University Jena

Monday 3rd November, 2014

Fibonacci Numbers

Fibonacci Numbers

Let $n \in \mathbb{N}$. The n-th Fibonacci number is defined by

$$F_n = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ F_{n-1} + F_{n-2} & \text{otherwise} \end{cases}$$

The first 10 Fibonacci numbers are 0, 1, 1, 2, 3, 5, 8, 13, 21, and 44.

From the definition we derive an algorithm that is obviously correct. But this algorithm is very inefficient (exponential in n).

Dynamic Programming

Observing that the same Fibonacci number is computed over and over again, we compute it once, store it, and look it up.

Solving sub problems only once and using the stored solution is called dynamic programming.

In our case the algorithm's complexity reduces to be linear in n while using linear memory in n.

Since only the last two Fibonacci numbers are accessed we can reduce the memory to be constant.

Closed Form Expression

We observed that

$$\begin{pmatrix} F_n \\ F_{n+1} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^n \begin{pmatrix} F_0 \\ F_1 \end{pmatrix}.$$

Evaluating this formula directly yields an algorithm with linear running time in n (due to the exponentiation).

Exponentiation by Squaring

By

$$x^n = \begin{cases} 1 & \text{if } n = 0 \\ x & \text{if } n = 1 \\ x^{2\frac{n}{2}} & \text{if } n \text{ is even} \\ x \cdot x^{2\frac{n-1}{2}} & \text{if } n \text{ is odd} \end{cases}$$

the n-th power can be reduced to logarithmic running time in n with exponentiation by squaring (n is halved in each step). This extends for $n \in \mathbb{Z}$ by using the inverse of x w.r.t. the multiplication operation, i.e., $x^{-n} = \frac{1}{x^n}$ for $x \in \mathbb{R}$.

Hence, we can compute F_n in $\mathcal{O}(\log n)$ time (assuming the multiplication takes constant time).

Simplifying the Formula

Eigenvalues

The eigenvalue λ with respect to the non-zero eigenvector x of a matrix A fulfills its defining equation

$$Ax = \lambda x.$$

Simplifying the Formula

Eigenvalues

The eigenvalue λ with respect to the non-zero eigenvector x of a matrix A fulfills its defining equation

$$Ax = \lambda x.$$

Note, that eigenvectors are invariant under scaling.

Simplifying the Formula

Eigenvalues

The eigenvalue λ with respect to the non-zero eigenvector x of a matrix A fulfills its defining equation

$$Ax = \lambda x.$$

Note, that eigenvectors are invariant under scaling.

To simplify our formula we compute eigenvalues/eigenvector of the matrix

$$F = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}.$$

Simplifying the Formula

Eigenvalues

The eigenvalue λ with respect to the non-zero eigenvector x of a matrix A fulfills its defining equation

$$Ax = \lambda x.$$

Note, that eigenvectors are invariant under scaling.

To simplify our formula we compute eigenvalues/eigenvector of the matrix

$$F = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}.$$

The eigenvalues λ of F can be found by solving $\det(F - \lambda \mathrm{I}) = 0$. We solve $-\lambda(1 - \lambda) - 1 = 0$ and get $\lambda_{0,1} = \frac{1 \pm \sqrt{5}}{2}$.

Simplifying the Formula

Eigenvectors

To obtain the corresponding eigenvectors for λ_0 we solve

$$(F - \lambda_0 \mathrm{I})\, x_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Simplifying the Formula

Eigenvectors

To obtain the corresponding eigenvectors for λ_0 we solve

$$(F - \lambda_0 \mathrm{I})\, x_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Since eigenvectors are invariant under scaling we set the first component to 1 and get

$$x_0 = \begin{pmatrix} 1 \\ \lambda_0 \end{pmatrix}.$$

Simplifying the Formula

Eigenvectors

To obtain the corresponding eigenvectors for λ_0 we solve

$$(F - \lambda_0 \mathrm{I})\, x_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Since eigenvectors are invariant under scaling we set the first component to 1 and get

$$x_0 = \begin{pmatrix} 1 \\ \lambda_0 \end{pmatrix}.$$

Analogously we get

$$x_1 = \begin{pmatrix} 1 \\ \lambda_1 \end{pmatrix}$$

for λ_1.

Simplifying the Formula

Represent $(F_0, F_1)'$ in x_0 and x_1

Since x_0 and x_1 are orthogonal we can express $\begin{pmatrix} F_0 \\ F_1 \end{pmatrix}$ as a linear combination of x_0 and x_1.

Simplifying the Formula

Represent $(F_0, F_1)'$ in x_0 and x_1

Since x_0 and x_1 are orthogonal we can express $\begin{pmatrix} F_0 \\ F_1 \end{pmatrix}$ as a linear combination of x_0 and x_1.

We solve

$$\begin{pmatrix} x_0 & x_1 \end{pmatrix} \alpha = \begin{pmatrix} F_0 \\ F_1 \end{pmatrix}$$

and get

$$\alpha = \begin{pmatrix} \frac{1}{\lambda_0 - \lambda_1} \\ \frac{-1}{\lambda_0 - \lambda_1} \end{pmatrix}.$$

Simplifying the Formula

Represent $(F_0, F_1)'$ in x_0 and x_1

Since x_0 and x_1 are orthogonal we can express $\begin{pmatrix} F_0 \\ F_1 \end{pmatrix}$ as a linear combination of x_0 and x_1.

We solve

$$\begin{pmatrix} x_0 & x_1 \end{pmatrix} \alpha = \begin{pmatrix} F_0 \\ F_1 \end{pmatrix}$$

and get

$$\alpha = \begin{pmatrix} \frac{1}{\lambda_0 - \lambda_1} \\ \frac{-1}{\lambda_0 - \lambda_1} \end{pmatrix}.$$

Since $\lambda_0 - \lambda_1 = \frac{1+\sqrt{5}}{2} - \frac{1-\sqrt{5}}{2} = \frac{1+\sqrt{5}-1+\sqrt{5}}{2} = \sqrt{5}$ we write further

$$\alpha = \begin{pmatrix} \frac{1}{\sqrt{5}} \\ \frac{-1}{\sqrt{5}} \end{pmatrix}.$$

Simplifying the Formula

We simplify our formula

$$F^n \begin{pmatrix} F_0 \\ F_1 \end{pmatrix} = F^n \left(\alpha_0 x_0 + \alpha_1 x_1\right)$$

Simplifying the Formula

We simplify our formula

$$
\begin{aligned}
F^n \begin{pmatrix} F_0 \\ F_1 \end{pmatrix} &= F^n \left(\alpha_0 x_0 + \alpha_1 x_1\right) \\
&= \alpha_0 F^n x_0 + \alpha_1 F^n x_1
\end{aligned}
$$

Simplifying the Formula

We simplify our formula

$$\begin{aligned} F^n \begin{pmatrix} F_0 \\ F_1 \end{pmatrix} &= F^n \left(\alpha_0 x_0 + \alpha_1 x_1\right) \\ &= \alpha_0 F^n x_0 + \alpha_1 F^n x_1 \\ &= \alpha_0 F^{n-1} \lambda_0 x_0 + \alpha_1 F^{n-1} \lambda_1 x_1 \end{aligned}$$

Simplifying the Formula

We simplify our formula

$$
\begin{aligned}
F^n \begin{pmatrix} F_0 \\ F_1 \end{pmatrix} &= F^n (\alpha_0 x_0 + \alpha_1 x_1) \\
&= \alpha_0 F^n x_0 + \alpha_1 F^n x_1 \\
&= \alpha_0 F^{n-1} \lambda_0 x_0 + \alpha_1 F^{n-1} \lambda_1 x_1 \\
&= \alpha_0 \lambda_0^n x_0 + \alpha_1 \lambda_1^n x_1
\end{aligned}
$$

Simplifying the Formula

We simplify our formula

$$\begin{aligned}
F^n \begin{pmatrix} F_0 \\ F_1 \end{pmatrix} &= F^n \left(\alpha_0 x_0 + \alpha_1 x_1\right) \\
&= \alpha_0 F^n x_0 + \alpha_1 F^n x_1 \\
&= \alpha_0 F^{n-1} \lambda_0 x_0 + \alpha_1 F^{n-1} \lambda_1 x_1 \\
&= \alpha_0 \lambda_0^n x_0 + \alpha_1 \lambda_1^n x_1 \\
&= \frac{1}{\sqrt{5}} \lambda_0^n \begin{pmatrix} 1 \\ \lambda_0 \end{pmatrix} - \frac{1}{\sqrt{5}} \lambda_1^n \begin{pmatrix} 1 \\ \lambda_1 \end{pmatrix}
\end{aligned}$$

Simplifying the Formula

We simplify our formula

$$
\begin{aligned}
F^n \begin{pmatrix} F_0 \\ F_1 \end{pmatrix} &= F^n \left(\alpha_0 x_0 + \alpha_1 x_1\right) \\
&= \alpha_0 F^n x_0 + \alpha_1 F^n x_1 \\
&= \alpha_0 F^{n-1} \lambda_0 x_0 + \alpha_1 F^{n-1} \lambda_1 x_1 \\
&= \alpha_0 \lambda_0^n x_0 + \alpha_1 \lambda_1^n x_1 \\
&= \frac{1}{\sqrt{5}} \lambda_0^n \begin{pmatrix} 1 \\ \lambda_0 \end{pmatrix} - \frac{1}{\sqrt{5}} \lambda_1^n \begin{pmatrix} 1 \\ \lambda_1 \end{pmatrix}
\end{aligned}
$$

which gives us

$$
F_n = \frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2} \right)^n - \frac{1}{\sqrt{5}} \left(\frac{1-\sqrt{5}}{2} \right)^n .
$$

Simplifying the Formula

By induction over $n \in \mathbb{N}$ we show

$$-\frac{1}{2} < \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n < \frac{1}{2}.$$

It holds for $n = 0$ and $n = 1$ and

$$\frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^{n+1} = \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n \underbrace{\frac{1-\sqrt{5}}{2}}_{>-\frac{1}{2} \text{ and } <\frac{1}{2}}$$

which proves our claim.

Simplified Formula

By leaving out the second term we make at most an error of $< \frac{1}{2}$ and get

$$F_n = \left\lfloor \frac{1}{\sqrt{5}} \left(\frac{1 + \sqrt{5}}{2} \right)^n + \frac{1}{2} \right\rfloor .$$

Trades

- Constants for variables/constants
- Time for space
- Compiling time/start up for running time
- Development time for running time