Week 0

Welcome To SIGma

SIGma

Outline of a Short Meeting

Officers in No Particular Order

Computing Fibonacci

Open Forum

Anakin (@Spamakin)

- Math Major
- SIGPwny Crypto[1] Gang + Admin team
- CA for CS 173 + 374
- Research with Sam
- Intern at CME Group over the summer

[1] Not that one, the other one

Sam (@Surg)

- Summer Amazon Intern
- CS Major
- Doing CS374 Course Dev
- Doing Theory Research with Sariel Har-Peled
- Research with Anakin

Husnain

- Math Major
- SIGPwny Crypto Gang + Helper
- Project Euler Enthusiast

Aditya (`@nebu`)

- ECE/Math double degree.
- Worked on fast Ethernet error correction hardware over the summer.
- Other interests: FP, PL, Crypto.

Hassam

- Intern at Amazon over the summer
- CS/Math Dual Major
- SIGPwny Crypto Gang + Admin team
- CA for CS 233
- Compiler research

Phil (`@fizzle`)

- CS/Ling Major
- CA for CS 233
- SIGecom - game theory, economics, and computation

Section 2

Computing Fibonacci

Recursive

$$F_{n+1} = F_n + F_{n-1} \quad (n \geq 1; F_0 = 0; F_1 = 1)$$

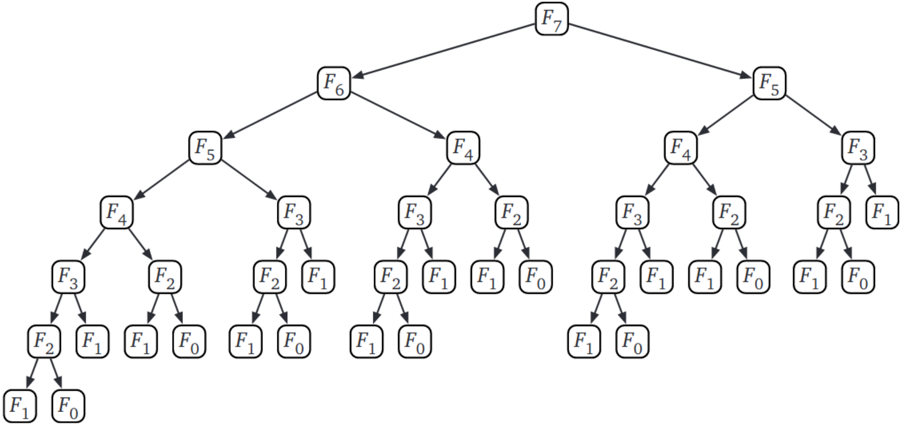

Figure: From Algorithms by Jeff Erickson

Iterative

FIBONACCI(n):

$prev, curr \leftarrow 1, 0$

for $i \leftarrow 1 \dots n$

$\quad next \leftarrow curr + prev$

$\quad prev \leftarrow curr$

$\quad curr \leftarrow next$

return $curr$

Aside: Square-and-Multiply

Before we get to the even faster way to compute F_n, we first look how to compute powers of a number quickly.
Say we want to compute x^8. We can use 7 multiplications as follows:

$$x \to x^2 \to x^3 \to x^4 \to x^5 \to x^6 \to x^7 \to x^8$$

But we can use just 3:

$$x \to x^2 \to x^4 \to x^8$$

.

We can use 12 multiplications to compute x^{13} as follows:

$$x \to x^2 \to x^3 \to x^4 \to x^5 \to x^6 \to x^7 \to x^8 \to x^9 \to x^{10} \to x^{11} \to x^{12} \to x^{13}$$

But if we first compute powers as such

$$x^2 \leftarrow x \cdot x$$
$$x^4 \leftarrow x^2 \cdot x^2$$
$$x^8 \leftarrow x^4 \cdot x^4$$

And then using these we get $x^8 \cdot x^4 \cdot x = x^{13}$ in just 6 total multiplications.
We can generalize this using binary

1	**1**	0	**1**

```
POWER(x, n):
    curr ← 1
    for i ← 1 ... n :
        curr ← curr * x
    return curr
```

```
SQUAREMULTPOWER(x, n):
    res, power ← 1, x
    for bit in BINARY(n):
        if bit = 1:
            res ← res * power
        power ← power * power
    return res
```

Matrices

We have the following two linear equations

$$\begin{aligned} F_n &= F_{n-1} + F_{n-2} \\ F_{n-1} &= F_{n-1} \end{aligned}$$

We can represent this as follows using matrices

$$\begin{bmatrix} F_{n-1} \\ F_n \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} F_{n-2} \\ F_{n-1} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}^2 \begin{bmatrix} F_{n-3} \\ F_{n-2} \end{bmatrix} = \ldots = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}^n \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Generating Functions

"A generating function is a clothesline on which we hang up a sequence of numbers for display"

— **Herbert Wilf**, *Generatingfunctionology*

Generating Functions

Explicitly, our recurrence is

$$F_{n+1} = F_n + F_{n-1} \quad (n \geq 1; F_0 = 0; F_1 = 1)$$

Define a "generating function": a function whose coefficients are the Fibonacci numbers

$$F(x) = \sum_{n \geq 0} F_n x^n = F_0 + F_1 x + F_2 x^2 + \cdots$$

Let's find this function!

Generating Functions: The LHS

Given,

$$F(x) = \sum_{n \geq 0} F_n x^n \qquad F_{n+1} = F_n + F_{n-1} \quad (n \geq 1)$$

Multiply LHS of recurrence by x^n, and take the sum for $n \geq 1$.

$$F_2 x + F_3 x^2 + F_4 x^3 + \cdots = \frac{F(x) - x}{x} \tag{1}$$

Generating Functions: The RHS

Given,

$$F(x) = \sum_{n \geq 0} F_n x^n \qquad F_{n+1} = F_n + F_{n-1} \quad (n \geq 1)$$

Multiply RHS of recurrence by x^n, and take the sum for $n \geq 1$.

$$\big(F_1 x + F_2 x^2 + F_3 x^3 + \cdots\big) + \big(F_0 x + F_1 x^2 + F_2 x^3 + \cdots\big)$$

$$= F(x) + xF(x) \tag{2}$$

Generating Functions: Equate LHS and RHS

$$\frac{F(x) - x}{x} = F(x) + x \cdot F(x)$$

$$\implies F(x) = \frac{x}{1 - x - x^2}$$

That's our generating function!

Generating Functions: Some Further Analysis

Remember partial fraction decomposition?

$$F(x) = \frac{x}{1 - x - x^2}$$

$$F(x) = \frac{x}{(1 - xr_+)(1 - xr_-)} = \frac{1}{r_+ - r_-}\left(\frac{1}{1 - xr_+} - \frac{1}{1 - xr_-}\right)$$

where

$$r_\pm = \frac{1 \pm \sqrt{5}}{2}$$

Generating Functions: Geometric Series

For a geometric series,

$$\sum_{n=0}^{\infty} ar^n = \frac{a}{1-r}$$

Using the geometric series sum formula,

$$F(x) = \frac{1}{\sqrt{5}} \left(\frac{1}{1 - xr_+} - \frac{1}{1 - xr_-} \right)$$

$$\implies F(x) = \frac{1}{\sqrt{5}} \left(\sum_{i=0}^{\infty} r_+^i x^i - \sum_{i=0}^{\infty} r_-^i x^i \right)$$

Generating Functions: Almost There

Writing this out to make it a bit more obvious,

$$F(x) = \frac{1}{\sqrt{5}} \left(\sum_{i=0}^{\infty} (r_+^i - r_-^i) x^i \right)$$

Doesn't this look a lot like a polynomial? The coefficients are the Fibonacci numbers we are after!

Generating Functions: Closed Form

Picking off coefficients from the geometric series, we see that

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

Since $|r_-/\sqrt{5}| < 0.5$ for all $n \geq 0$, we can actually neglect it altogether to get a simpler closed form:

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

Summary

Algorithm	Time Complexity
Naive recursive	$O\left(2^n\right)$
Iterative	$O(n)$
Matrix	$O(\log n)$
Generating functions	$O(1)$

Section 3

Open Forum

Time?

Book?

Research?

So long, and thanks for all the fish!