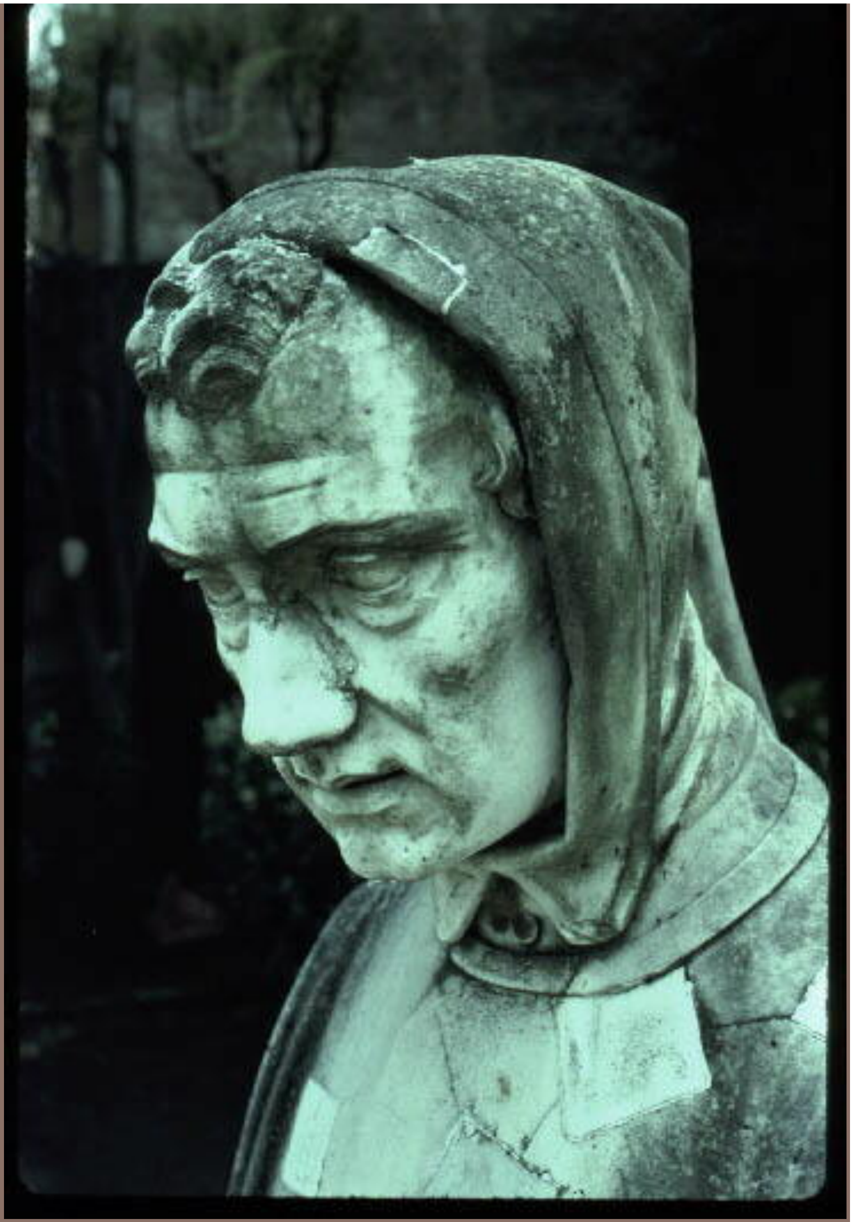
Fibonacci
(Leonardo Pisano)
1170-1240?
Statue in Pisa Italy

And recurrences

# FIBONACCI



Lecture 23
CS2110 – Spring 2015

# Fibonacci function (1202)

fib(0) = 0

fib(1) = 1

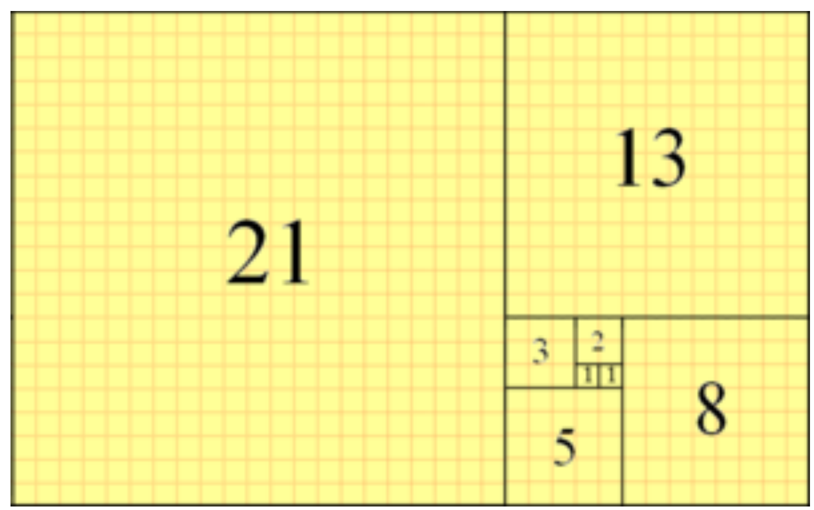
fib(n) = fib(n-1) + fib(n-2)  for $n \geq 2$

But sequence described much earlier in India

```
/** Return fib(n). Precondition: n ≥ 2.*/
public static int f(int n) {
    if ( n <= 1) return n;
    return f(n-1) + f(n-2);
}
```

0, 1, 1, 2, 3, 5, 8, 13, 21, …

Downloaded from wikipedia

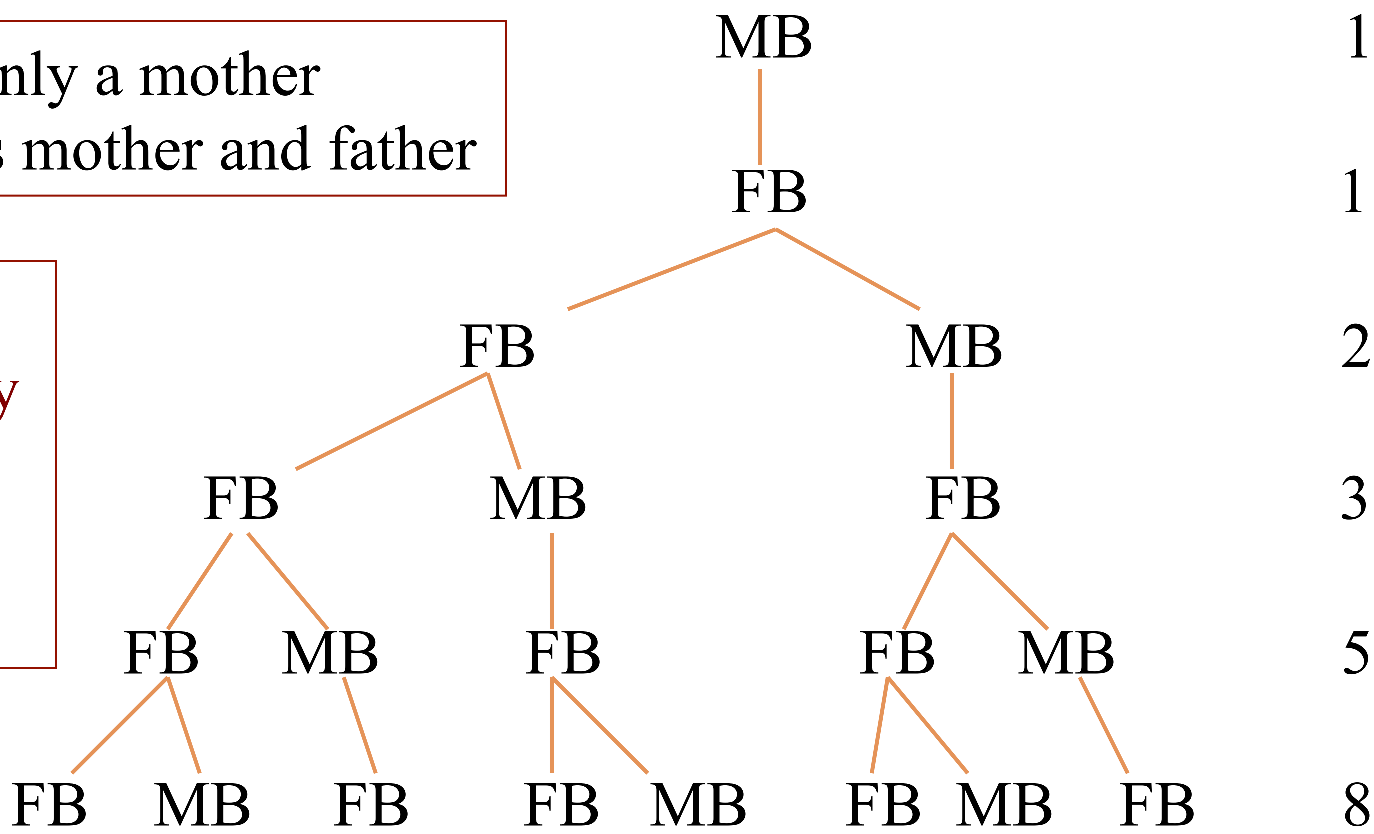


# fibonacci and bees



Male bee has only a mother
Female bee has mother and father

The number of ancestors at any level is a Fibonnaci number




MB: male bee, FB: female bee

# Fibonacci in nature



The artichoke uses the Fibonacci pattern to spiral the sprouts of its flowers.

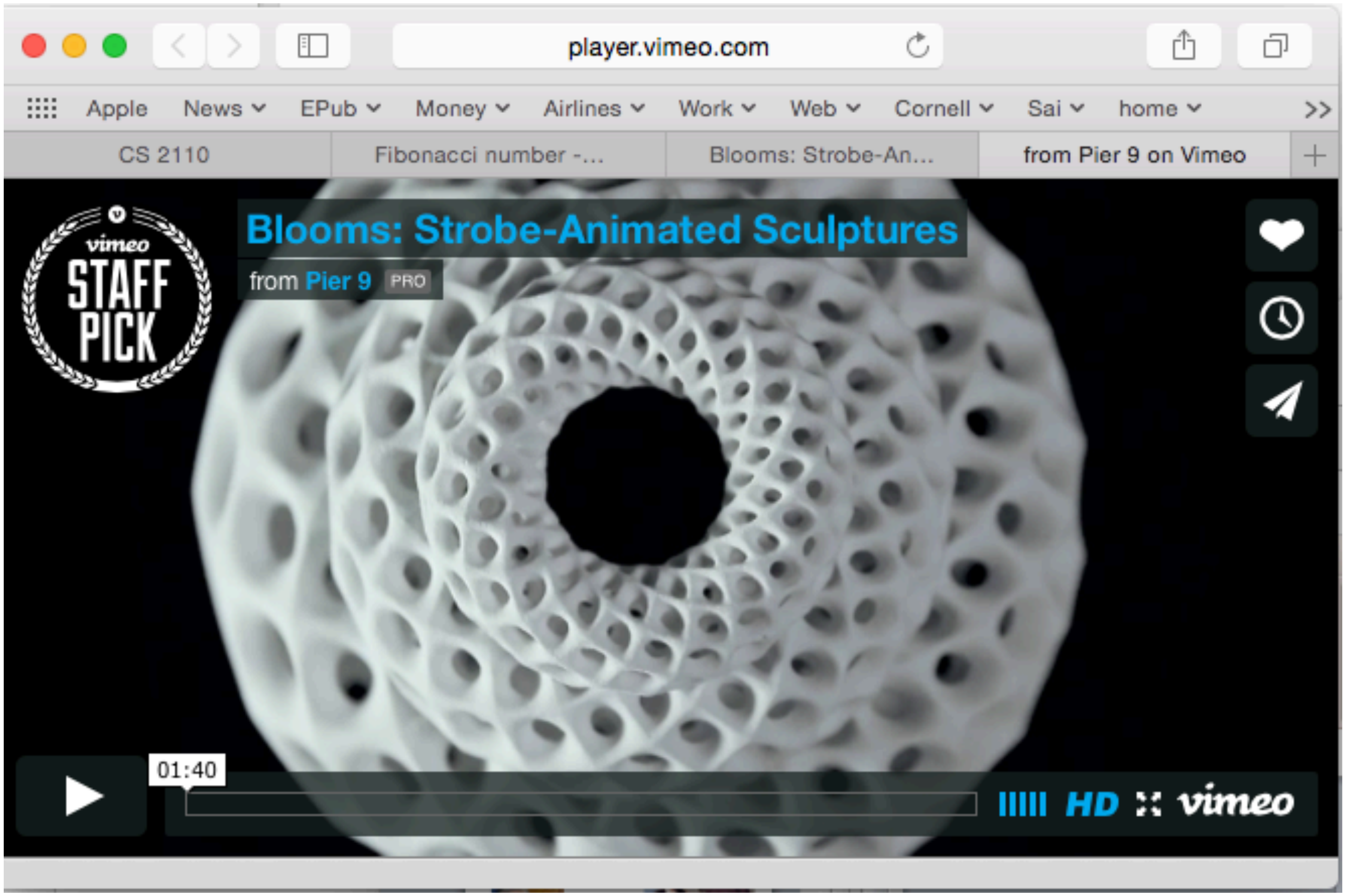
The artichoke sprouts its leafs at a constant amount of rotation: 222.5 degrees (in other words the distance between one leaf and the next is 222.5 degrees). You can measure this rotation by dividing 360 degrees (a full spin) by the inverse of the golden ratio. We see later how the golden ratio is connected to fibonacci.

topones.weebly.com/1/post/2012/10/the-artichoke-and-fibonacci.html

# Blooms: strobe-animated sculptures



www.instructables.com/id/Blooming-Zoetrope-Sculptures/



# Uses of Fibonacci sequence in CS

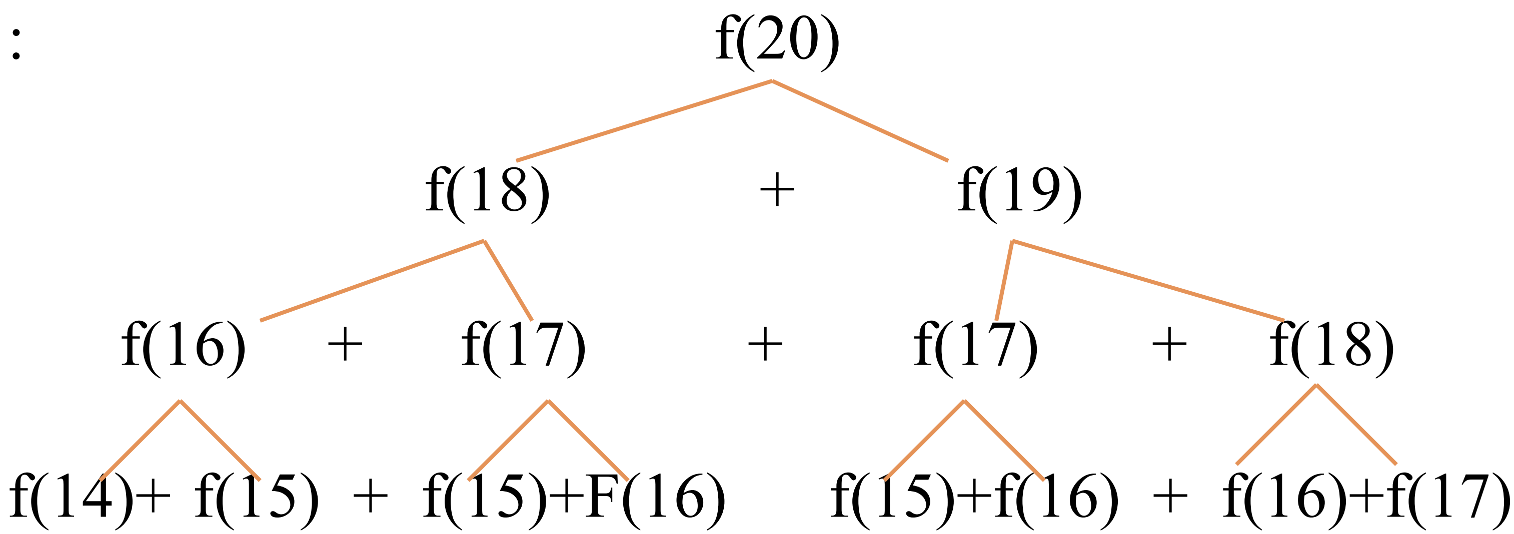


- Fibonacci search
- Fibonacci heap data strcture
- Fibonacci cubes: graphs used for interconnecting parallel and distributed systems

# Recursion for fib: f(n) = f(n-1) + f(n-2)



:




Calculates f(15) four times! What is complexity of f(n)?

# Recursion for fib: $f(n) = f(n-1) + f(n-2)$



$T(0) = a$ Recurrence relation for the time

$T(1) = a$

$T(n) = T(n-1) + T(n-2)$

Theorem: $T(n)$ is $O(2^n)$

Theorem: $P(n)$ holds for $n >= N$:

$P(n)$: $T(n) \leq c\ 2^n$

Base case: $T(0) = a \leq c\ 2^0$ (use $c = a$ or anything bigger)

Base case: $T(1) = a \leq c\ 2^1$ (use $c = a$ or anything bigger)

# Recursion for fib: f(n) = f(n-1) + f(n-2)



Recurrence relation for the time

$T(0) = a$

$T(1) = a$

$T(n) = T(n-1) + T(n-2) + a$

Theorem: P(n) holds for $n >= N$:

$P(n): T(n) \le c\, 2^n$

Inductive case: Assume $P(0), \ldots, P(k)$, $k \ge 2$, and prove $P(k+1)$

$T(k+1)$

$=$ <def of T>

$T(k) + T(k-1) + a$

$<=$ <P(k), P(k-1)>

$c\, 2^k + c\, 2^{k-1} + a$

<arith>

$= c\, 2^{k+1}(1/2 + 1/4) + a$

<arith, choose c = a>

$\le c\, 2^{k+1}$

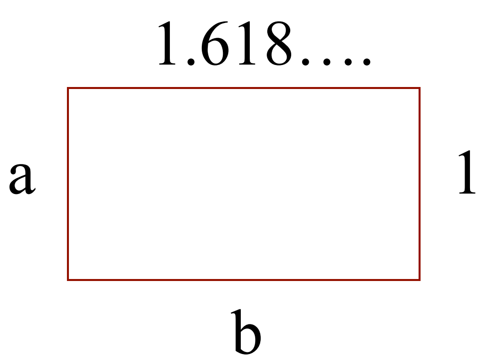
# The golden ratio



$a > 0$ and $b > a > 0$ are in the **golden ratio** if

$(a + b) / a = a/b$ call that value $\varphi$

$\varphi^2 = \varphi + 1$ so $\varphi = (1 + \text{sqrt}(5)) / 2 = 1.618 \ldots$



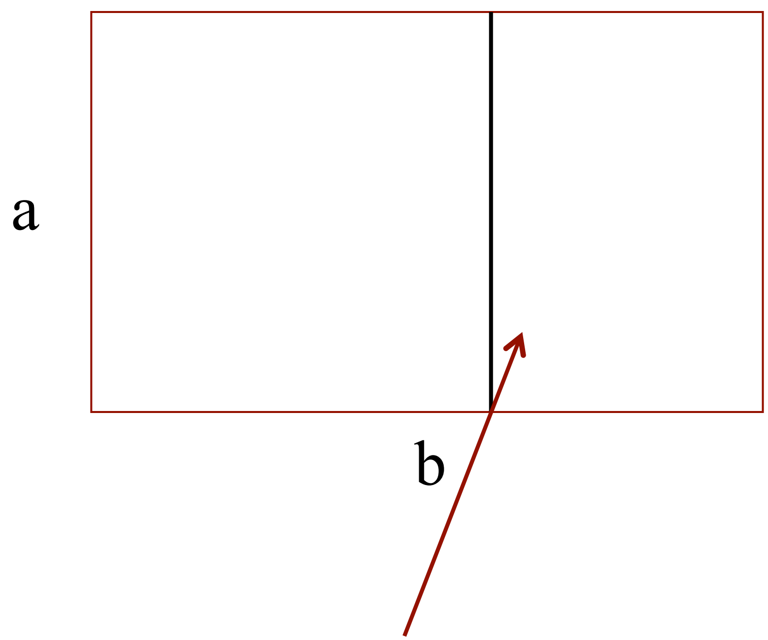

ratio of sum of sides to longer side

=

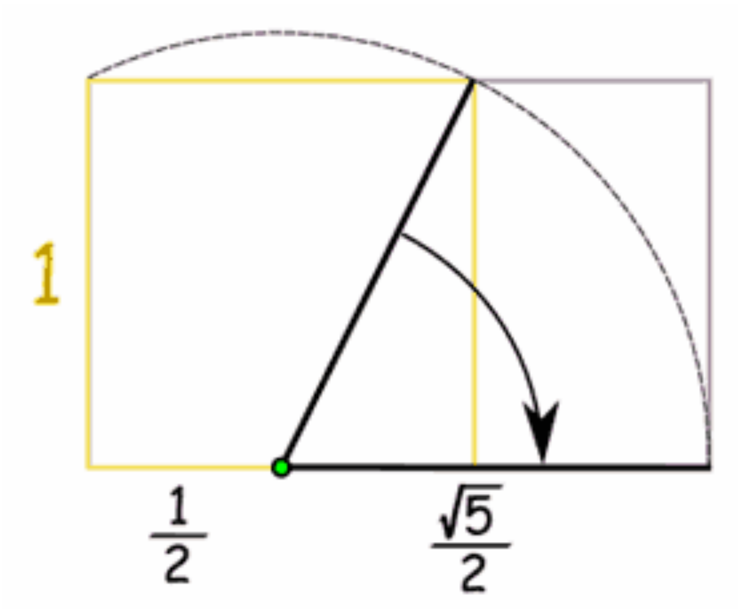
ratio of longer side to shorter side

# The golden ratio






golden rectangle




How to draw a golden rectangle

# The Parthenon



# Can prove that Fibonacci recurrence is $O(\varphi^n)$



We won't prove it.

Requires proof by induction

Relies on identity $\varphi^2 = \varphi + 1$

# Linear algorithm to calculate fib(n)



```
/** Return fib(n), for n >= 0. */
public static int f(int n) {
   if (n <= 1) return 1;
   int p= 0;   int c= 1;  int i= 2;
   // invariant: p = fib(i-2) and c = fib(i-1)
  while (i < n) {
        int fibi= c + p;   p= c;  c= fibi;
        i= i+1;
   }
   return c + p;
}
```

# Logarithmic algorithm!



$f_0 = 0$

$f_1 = 1$

$f_{n+2} = f_{n+1} + f_n$

You know a logarithmic algorithm for exponentiation —recursive and iterative versions

$$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} f_n \\ f_{n+1} \end{pmatrix} = \begin{pmatrix} f_{n+1} \\ f_{n+2} \end{pmatrix} \quad \text{so:} \quad \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^n \begin{pmatrix} f_0 \\ f_1 \end{pmatrix} = \begin{pmatrix} f_n \\ f_{n+1} \end{pmatrix}$$

Gries and Levin/ Computing a Fibonacci number in log time. IPL 2 (October 1980), 68-69.

# Constant-time algorithm!



Define **$\phi = (1 + \sqrt{5}) / 2$** **$\phi' = (1 - \sqrt{5}) / 2$**

The golden ratio again.

Prove by induction on n that

$$fn = (\phi^n - \phi'^n) / \sqrt{5}$$

We went from $O(2^n)$ to $O(1)$