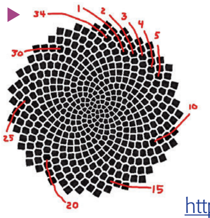
# Fibonacci Numbers



https://www.mathsisfun.com/numbers/fibonacci-sequence.html

**Quiz**
What is $F(13)$?

| n | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0, | 1, | 1, | 2, | 3, | 5, | 8, | 13, | 21, | 34, | 55, | 89, | 144, ... |

$$0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, \cdots$$

**Quiz**
How many ways to specify a sequence? Which ones are shown? What's missing?

$$F(n) = F(n-1) + F(n-2), \quad n > 1$$

$$F(0) = 0, F(1) = 1$$

**Quiz**
What is $F(15)$? See Exercise in Slide 5.

$$\boxed{F(n) - F(n-1) - F(n-2) = 0, \quad n > 1}$$

?

## Recognize form?

# Standard Recurrence
# Linear 2nd Order

App. B

- **Constant coefficients**
- **Homogeneous case**
- **Characteristic equation**

$$ax(n) + bx(n-1) + cx(n-2) = f(n)$$

$$f(n) = 0 \;\Rightarrow\; ax(n) + bx(n-1) + cx(n-2) = 0$$

A linear, 2nd order, **homogeneous** recurrence with constant coefficients.

$$ar^2 + br + c = 0$$

$$x(n) = \begin{cases} \alpha r_1^n + \beta r_2^n & r_1,\ r_2 \text{ real and distinct} \\ \alpha r^n + \beta n r^n & r_1 = r_2 = r \\ \gamma^n[\alpha \cos n\theta + \beta \sin n\theta] & r_{1,2} = u \pm iv \text{ distinct complex} \end{cases}$$

where $\gamma = \sqrt{u^2 + v^2}, \quad \theta = \arctan v/u$

# Fibonacci Sequence
# Solve Recurrence

**Reminder**

$$ax^2 + bx + c = 0$$

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

**Quiz**
**Name the formula. What are the values of $a$,$b$,$c$ in this case?**

$$r^2 - r - 1 = 0$$

**Quiz**
**Which case of the recurrence solution applies?**

WolframAlpha

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

$$= \frac{-(-1) \pm \sqrt{(-1)^2 - 4(1)(-1)}}{2(1)}$$

$$F(n) = \alpha \left( \frac{1 + \sqrt{5}}{2} \right)^n + \beta \left( \frac{1 - \sqrt{5}}{2} \right)^n$$

## Solve Recurrence

# Determine Constants

The constants only differ in sign.

## Use recurrence initial conditions

$$F(0) = \alpha + \beta = 0$$

$$F(1) = \left(\frac{1+\sqrt{5}}{2}\right)\alpha + \left(\frac{1-\sqrt{5}}{2}\right)\beta = 1$$

**Exercise** WolframAlpha
Solve the system of equations (try by hand also).

## Solve for alpha and beta

$$\alpha = \frac{1}{\sqrt{5}}, \quad \beta = -\frac{1}{\sqrt{5}}$$

# Fibonacci Sequence
# The Generic Term

## ➪ Golden ratio

$\phi$ is an interesting special number called the **golden ratio** (values $\phi$, $\phi$-hat calculated earlier), check culture, history and myth.

https://www.goldennumber.net/parthenon-phi-golden-ratio
https://history.howstuffworks.com/history-vs-myth/parthenon-golden-ratio.htm

**Let** $$\phi = \frac{1+\sqrt{5}}{2}, \quad \hat{\phi} = \frac{1-\sqrt{5}}{2}$$

$$F(n) = \frac{1}{\sqrt{5}}(\phi^n - \hat{\phi}^n)$$

$\hat{\phi} = 1-\phi$

**Exercise**
Use approximation formula to compute $F(15)$. Ans. last slide.

## Approximation formula

# Fibonacci Sequence
# Computing Terms

**Quiz**
**Can be written directly from definition, but is it better than a straight programming-101 iterative version?**

**Algorithm** $F(n)$

```
if n ≤ 1 then
    return n
else return F(n − 1) + F(n − 2)
```

**Algorithm** $Fib(n)$

```
F[0] ← 0; F[1] ← 1
for i ← 2 to n do
    F[i] ← F[i − 1] + F[i − 2]
return F[n]
```

## Performance

## Computing Terms
# Recursion Efficiency

*B*(*n*) generates the same terms as *F*(*n*) but in a sequence that runs a step ahead, i.e., another particular solution of the same recurrence.

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0, | 1, | 1, | 2, | 3, | 5, | 8, | 13, | 21, | **34**, | 55, | 89, | 144, ··· |
| $B(n)$ ▶ | 1, | 1, | 2, | 3, | 5, | 8, | 13, | 21, | **34**, | 55, | 89, | 144, ··· | |

$$A(n) = B(n) - 1$$
$$= F(n+1) - 1$$
$$= \frac{1}{\sqrt{5}}(\phi^{n+1} - \hat{\phi}^{n+1}) - 1$$

**Quiz**
How many additions are needed to compute *F*(8)? (Check shown sequences).

**Exercise**
Use code to verify answer (code the recursive version, insert code to count additions, run for *n*=8). Output *A*(*n*), verify cases *n*=0,1,2 both from code and pseudocode.

**Exercise** WolframAlpha
Try `fibonacci(100)`.

# How bad is it?

# Summary

**Exercise**
**Verify using *n*=4 that the recursive algorithm indeed does compute the same Fibonacci numbers repeatedly (redundantly).**

**Exercise**
Compare efficiency when input size is expressed as <u>number of bits</u> in *n*?

$$\frac{\phi^n}{\sqrt{5}}$$

**Quiz**
How would you calculate *F*(100)? Compare efficiency of generating sequence vs. one term.

## ➪ Use definition (recurrence)

- ✎ Iteratively: linear $n$
- ✎ Recursively: exp. $n$ (why?)

## ➪ Use approximation

Exponentiation, how efficient?

*1.61803^15 ÷ √‾5 in a hand calculator yields 609.977... F(15) = 610 (note an extra digit in ϕ beyond the 0 is needed when raising to relatively high power).*