# Discrete Structures: Recursion

Amotz Bar-Noy

Department of Computer and Information Science
Brooklyn College

# Recursion

## Poem

- Great fleas have little fleas upon their backs to bite 'em, And little fleas have lesser fleas, and so ad infinitum. And the great fleas themselves in turn have greater fleas to go on; While these again have greater still, and greater still, and so on.

## Illustrations

- https://storage.googleapis.com/algodailyrandomassets/curriculum/recursion/cover.jpg
- https://upload.wikimedia.org/wikipedia/commons/thumb/8/80/SierpinskiTriangle.svg/887px-SierpinskiTriangle.svg.png
- https://theburningmonk.com/wp-content/uploads/2017/08/recursion-01.png

# Recursion

## Definition

- Recursion occurs when something is defined in terms of its type.

## Focus

- **Recursive formulas** in mathematics.
- **Recursive programs** in computer science.

# Recursive Formulas

### Definition

- A recursive formula is usually defined on the set of integers greater than or equal to some number $m$ (usually 0 or 1).
- The formula computes the $n^{\text{th}}$ value based on some or all of the previous $n-1$ values.

### Goal

- Given initial values and a recursive formula, find an equivalent **closed-form expression** as a function of $n$ that does not depend on previous values.

### Recursion and induction

- Usually proving the correctness of a **solution** (a closed-form expression) to a recursive formula is done by induction.

# The Non-Negative Integers

**The recursive formula**

$$N(n) = \begin{cases} 0 & \text{for } n = 0 \\ N(n-1) + 1 & \text{for } n > 0 \end{cases}$$

**The recursive pseudocode**

**function** $N(n)$ (* integer $n \geq 0$ *)
  **if** $n = 0$
  **then return** $(0)$
  **else return** $(N(n-1) + 1)$

# The Non-Negative Integers

## Top-Down evaluation

$$\begin{aligned}
N(n) &= (N(n-1)+1) \\
&= (N(n-2)+1)+1 \\
&= (N(n-3)+1)+1+1 \\
&= (N(n-4)+1)+1+1+1 \\
&\vdots \\
&= (N(n-n)+1)+1+\cdots+1 \\
&= N(0)\overbrace{+1+1+\cdots+1}^{n} \\
&= 0+n \\
&= n
\end{aligned}$$

# The Non-Negative Integers

## The closed-form expression

$$N(n) = n$$

## Proof by induction

- Induction base: $N(0) = 0$
- Induction hypothesis: $N(n-1) = n - 1$ for $n > 0$
- Inductive step for $n > 0$:

$$\begin{aligned} N(n) &= N(n-1) + 1 \\ &= (n-1) + 1 \\ &= n \end{aligned}$$

# The Non-Negative Integers

## Bottom-Up evaluation

$$\begin{array}{rclclcl}
N(0) & = & 0 & = & 0 & = & 0 \\
N(1) & = & N(0)+1 & = & 0+1 & = & 1 \\
N(2) & = & N(1)+1 & = & 1+1 & = & 2 \\
N(3) & = & N(2)+1 & = & 2+1 & = & 3 \\
N(4) & = & N(3)+1 & = & 3+1 & = & 4 \\
 & \vdots & & \vdots & & \vdots & \\
N(n) & = & N(n-1)+1 & = & (n-1)+1 & = & n
\end{array}$$

# The Non-Negative Integers

## The non-recursive pseudocode

**function** $N(n)$ (* integer $n \geq 0$ *)
    $k = 0$
    **for** $i = 1$ **to** $n$
        $k = k + 1$
    **return**$(k)$

## The two pseudocodes are equivlent

- The **recursive pseudocode** implements the **Top-Down** evaluation.
- The **non-recursive pseudocode** implements the **Bottom-Up** evaluation.

# The Non-Positive Integers

### The recursive formula

$$N(n) = \begin{cases} 0 & \text{for } n = 0 \\ N(n-1) - 1 & \text{for } n > 0 \end{cases}$$

### The recursive pseudocode

**function** $N(n)$ (* integer $n \geq 0$ *)
  **if** $n = 0$
  **then return** $(0)$
  **else return** $(N(n-1) - 1)$

# The Non-Positive Integers

## Top-Down evaluation

$$
\begin{aligned}
N(n) &= (N(n-1)-1) \\
&= (N(n-2)-1)-1 \\
&= (N(n-3)-1)-1-1 \\
&= (N(n-4)-1)-1-1-1 \\
&\vdots \\
&= (N(n-n)-1)-1-\cdots-1 \\
&= N(0)\overbrace{-1-1-\cdots-1}^{n} \\
&= 0-n \\
&= -n
\end{aligned}
$$

# The Non-Positive Integers

### The closed-form expression

$$N(n) = -n$$

### Proof by induction

- Induction base: $N(0) = 0 = -0$
- Induction hypothesis: $N(n-1) = -(n-1)$ for $n > 0$
- Inductive step for $n > 0$:

$$\begin{aligned} N(n) &= N(n-1) - 1 \\ &= -(n-1) - 1 \\ &= -n \end{aligned}$$

# The Non-Positive Integers

## Bottom-Up evaluation

$$
\begin{array}{rcccccc}
N(0) & = & 0 & = & -0 & = & -0 \\
N(1) & = & N(0) - 1 & = & -0 - 1 & = & -1 \\
N(2) & = & N(1) - 1 & = & -1 - 1 & = & -2 \\
N(3) & = & N(2) - 1 & = & -2 - 1 & = & -3 \\
N(4) & = & N(3) - 1 & = & -3 - 1 & = & -4 \\
 & \vdots & & \vdots & & \vdots & \\
N(n) & = & N(n-1) - 1 & = & -(n-1) - 1 & = & -n
\end{array}
$$

# The Non-Positive Integers

## The non-recursive pseudocode

**function** $N(n)$ (* integer $n \geq 0$ *)
  $k = -0$
  **for** $i = 1$ **to** $n$
    $k = k - 1$
  **return**$(k)$

## The two pseudocodes are equivlent

- The **recursive pseudocode** implements the **Top-Down** evaluation.
- The **non-recursive pseudocode** implements the **Bottom-Up** evaluation.

# Another Recursive Formula

## The recursive formula

$$T(n) = \begin{cases} 2 & \text{for } n = 0 \\ T(n-1) + 10 & \text{for } n > 0 \end{cases}$$

## The recursive pseudocode

**function** $T(n)$ (* integer $n \geq 0$ *)
  **if** $n = 0$
  **then return** $(2)$
  **else return** $(T(n-1) + 10)$

# Another Recursive Formula

## Top-Down evaluation

$$
\begin{aligned}
T(n) &= (T(n-1)+10) \\
&= (T(n-2)+10)+10 \\
&= (T(n-3)+10)+10+10 \\
&= (T(n-4)+10)+10+10+10 \\
&\vdots \\
&= (T(n-n)+10)+10+\cdots+10 \\
&= T(0)\overbrace{+10+10+\cdots+10}^{n} \\
&= 2+10n
\end{aligned}
$$

# Another Recursive Formula

## The closed-form expression

$$T(n) = 2 + 10n$$

## Proof by induction

- Induction base: $T(0) = 2 + 10 \cdot 0 = 2$
- Induction hypothesis: $T(n-1) = 2 + 10(n-1)$ for $n > 0$
- Inductive step for $n > 0$:

$$\begin{aligned} T(n) &= T(n-1) + 10 \\ &= (2 + 10(n-1)) + 10 \\ &= 2 + 10n - 10 + 10 \\ &= 2 + 10n \end{aligned}$$

# Another Recursive Formula

## Bottom-Up evaluation

$$
\begin{array}{rcccccc}
T(0) & = & 2 & = & 2 & = & 2 \\
T(1) & = & T(0)+10 & = & 2+10 & = & 12 \\
T(2) & = & T(1)+10 & = & 12+10 & = & 22 \\
T(3) & = & T(2)+10 & = & 22+10 & = & 32 \\
T(4) & = & T(3)+10 & = & 32+10 & = & 42 \\
 & \vdots & & \vdots & & \vdots & \\
T(n) & = & T(n-1)+10 & = & 2+10(n-1)+10 & = & 2+10n
\end{array}
$$

# Another Recursive Formula

**The non-recursive pseudocode**

```
function T(n)   (* integer n ≥ 0 *)
   t = 2
   for i = 1 to n
      t = t + 10
   return(t)
```

**The two pseudocodes are equivlent**

- The **recursive pseudocode** implements the **Top-Down** evaluation.
- The **non-recursive pseudocode** implements the **Bottom-Up** evaluation.

# Arithmetic Progressions

## The recursive formula

$$A(n) = \begin{cases} a_0 & \text{for } n = 0 \text{ and a real number } a_0 \\ A(n-1) + d & \text{for } n > 0 \text{ and a real number } d \end{cases}$$

## The recursive pseudocode

**function** $A(n)$ (* integer $n \geq 0$ *)
  **if** $n = 0$
  **then return** $(a_0)$
  **else return** $(A(n-1) + d)$

# Arithmetic Progressions

## Top-Down evaluation

$$
\begin{aligned}
A(n) &= (A(n-1)+d) \\
&= (A(n-2)+d)+d \\
&= (A(n-3)+d)+d+d \\
&= (A(n-4)+d)+d+d+d \\
&\vdots \\
&= (A(n-n)+d)+d+\cdots+d \\
&= A(0)\overbrace{+d+d+\cdots+d}^{n} \\
&= a_0+dn
\end{aligned}
$$

# Arithmetic Progressions

## The closed-form expression

$$A(n) = a_0 + dn$$

## Proof by induction

- Induction base: $A(0) = a_0 = a_0 + d \cdot 0$
- Induction hypothesis: $A(n-1) = a_0 + d(n-1)$ for $n > 0$
- Inductive step for $n > 0$:

$$\begin{aligned} A(n) &= A(n-1) + d \\ &= a_0 + d(n-1) + d \\ &= a_0 + dn - d + d \\ &= a_0 + dn \end{aligned}$$

# Arithmetic Progressions

## Bottom-Up evaluation

$$
\begin{array}{rcccccc}
A(0) & = & a_0 & = & a_0 & = & a_0 + d \cdot 0 \\
A(1) & = & A(0) + d & = & (a_0 + d \cdot 0) + d & = & a_0 + d \cdot 1 \\
A(2) & = & A(1) + d & = & (a_0 + d \cdot 1) + d & = & a_0 + d \cdot 2 \\
A(3) & = & A(2) + d & = & (a_0 + d \cdot 2) + d & = & a_0 + d \cdot 3 \\
A(4) & = & A(3) + d & = & (a_0 + d \cdot 3) + d & = & a_0 + d \cdot 4 \\
 & \vdots & & \vdots & & \vdots & \\
A(n) & = & A(n-1) + d & = & (a_0 + d(n-1)) + d & = & a_0 + dn
\end{array}
$$

# Arithmetic Progressions

### The non-recursive pseudocode

**function** $A(n)$ (* integer $n \geq 0$ and reals $d$ and $a_0$ *)
  $a = a_0$
  **for** $i = 1$ **to** $n$
    $a = a + d$
  **return**$(a)$

### The two pseudocodes are equivlent

- The **recursive pseudocode** implements the **Top-Down** evaluation.
- The **non-recursive pseudocode** implements the **Bottom-Up** evaluation.

# Arithmetic Progressions

### The recursive formula

$$A(n) = \begin{cases} a_0 & \text{for } n = 0 \text{ and a real number } a_0 \\ A(n-1) + d & \text{for } n > 0 \text{ and a real number } d \end{cases}$$

### The closed-form expression

$$A(n) = a_0 + dn$$

### Special cases

- **Non-negative integers:** $a_0 = 0 \wedge d = 1 \implies A(n) = n$
- **Non-positive integers:** $a_0 = 0 \wedge d = -1 \implies A(n) = -n$
- **Non-negative even integers:** $a_0 = 0 \wedge d = 2 \implies A(n) = 2n$
- **Positive odd integers:** $a_0 = 1 \wedge d = 2 \implies A(n) = 2n + 1$

# Powers of Two

## The recursive formula

$$P(n) = \begin{cases} 1 & \text{for } n = 0 \\ 2P(n-1) & \text{for } n > 0 \end{cases}$$

## The recursive pseudocode

**function** $P(n)$ (* integer $n \geq 0$ *)
  **if** $n = 0$
  **then return** $(1)$
  **else return** $(2P(n-1))$

## The non-recursive pseudocode

**function** $P(n)$ (* integer $n \geq 0$ *)
  $p = 1$
  **for** $i = 1$ **to** $n$
    $p = 2 \cdot p$
  **return**$(p)$

# Powers of Two

## Top-Down evaluation

$$
\begin{aligned}
P(n) &= 2P(n-1) &&= 2^1 P(n-1) \\
&= 2^1(2P(n-2)) &&= 2^2 P(n-2) \\
&= 2^2(2P(n-3)) &&= 2^3 P(n-3) \\
&= 2^3(2P(n-4)) &&= 2^4 P(n-4) \\
&\vdots \\
&= 2^{n-1}(2P(n-n)) &&= 2^n P(n-n) \\
&= 2^n P(0) \\
&= 2^n
\end{aligned}
$$

# Powers of Two

## Bottom-Up evaluation

$$
\begin{array}{lclclcl}
P(0) & = & 1 & = & 1 & = & 2^0 \\
P(1) & = & 2P(0) & = & 2 \cdot 2^0 & = & 2^1 \\
P(2) & = & 2P(1) & = & 2 \cdot 2^1 & = & 2^2 \\
P(3) & = & 2P(2) & = & 2 \cdot 2^2 & = & 2^3 \\
P(4) & = & 2P(3) & = & 2 \cdot 2^3 & = & 2^4 \\
 & \vdots & & \vdots & & \vdots & \\
P(n) & = & 2P(n-1) & = & 2 \cdot 2^{n-1} & = & 2^n
\end{array}
$$

# Powers of Two

## The closed-form expression

$$P(n) = 2^n$$

## Proof by induction

- Induction base: $P(0) = 1 = 2^0$
- Induction hypothesis: $P(n-1) = 2^{n-1}$ for $n > 0$
- Inductive step for $n > 0$:

$$\begin{aligned} P(n) &= 2P(n-1) \\ &= 2 \cdot 2^{n-1} \\ &= 2^n \end{aligned}$$

# Factorials

### The recursive formula

$$F(n) = \begin{cases} 1 & \text{for } n = 1 \\ n \cdot F(n-1) & \text{for } n > 1 \end{cases}$$

### The recursive pseudocode

**function** $F(n)$ (* integer $n \geq 1$ *)
  **if** $n = 1$
  **then return** $(1)$
  **else return** $(n \cdot F(n-1))$

### The non-recursive pseudocode

**function** $F(n)$ (* integer $n \geq 1$ *)
  $f = 1$
  **for** $i = 2$ **to** $n$
    $f = i \cdot f$
  **return**$(f)$

# Factorials

## Top-Down evaluation

$$
\begin{aligned}
F(n) &= nF(n-1) \\
&= n(n-1)F(n-2) \\
&= n(n-1)(n-2)F(n-3) \\
&= n(n-1)(n-2)(n-3)F(n-4) \\
&\vdots \\
&= n(n-1)(n-2)(n-3)\cdots(n-(n-2))\cdot F(1) \\
&= n(n-1)(n-2)(n-3)\cdots 2\cdot 1 \\
&= n!
\end{aligned}
$$

# Factorials

## Bottom-Up evaluation

$$
\begin{array}{rcccccccc}
F(1) & = & & & & & 1 & = & 1! \\
F(2) & = & 2 \cdot F(1) & = & 2 \cdot 1 & = & 2 & = & 2! \\
F(3) & = & 3 \cdot F(2) & = & 3 \cdot 2 & = & 6 & = & 3! \\
F(4) & = & 4 \cdot F(3) & = & 4 \cdot 6 & = & 24 & = & 4! \\
F(5) & = & 5 \cdot F(4) & = & 5 \cdot 24 & = & 120 & = & 5! \\
 & \vdots & & & & & & \vdots & \\
F(n) & = & & & & & & = & n!
\end{array}
$$

# Factorials

## The closed-form expression

$$F(n) = n!$$

## Proof by induction

- Induction base: $F(1) = 1 = 1!$
- Induction hypothesis: $F(n-1) = (n-1)!$ for $n > 1$
- Inductive step for $n > 1$:

$$\begin{aligned} F(n) &= nF(n-1) \\ &= n(n-1)! \\ &= n! \end{aligned}$$

# The Sum $1 + 2 + \cdots + n$

## The recursive formula

$$S(n) = \begin{cases} 1 & \text{for } n = 1 \\ S(n-1) + n & \text{for } n > 1 \end{cases}$$

## The recursive pseudocode

**function** $S(n)$ (* integer $n \geq 1$ *)
  **if** $n = 1$
  **then return** $(1)$
  **else return** $(S(n-1) + n)$

## The non-recursive pseudocode

**function** $S(n)$ (* integer $n \geq 1$ *)
  $s = 1$
  **for** $i = 2$ **to** $n$
    $s = s + i$
  **return**$(s)$

# The Sum $1 + 2 + \cdots + n$

## Bottom-Up evaluation

- $S(1) = \quad = 1 = \frac{1 \cdot 2}{2}$
- $S(2) = S(1) + 2 = 1 + 2 = 3 = \frac{2 \cdot 3}{2}$
- $S(3) = S(2) + 3 = 3 + 3 = 6 = \frac{3 \cdot 4}{2}$
- $S(4) = S(3) + 4 = 6 + 4 = 10 = \frac{4 \cdot 5}{2}$
- $S(5) = S(4) + 5 = 10 + 5 = 15 = \frac{5 \cdot 6}{2}$
- $S(6) = S(5) + 6 = 15 + 6 = 21 = \frac{6 \cdot 7}{2}$
- $S(7) = S(6) + 4 = 21 + 7 = 28 = \frac{7 \cdot 8}{2}$

# The Sum $1 + 2 + \cdots + n$

**The closed-form expression**

$$S(n) = \frac{n(n+1)}{2}$$

**Proof by induction**

- Induction base: $S(1) = 1 = \frac{1 \cdot 2}{2}$
- Induction hypothesis: $S(n-1) = \frac{(n-1)n}{2}$ for $n > 1$
- Inductive step for $n > 1$:

$$\begin{aligned} S(n) &= S(n-1) + n \\ &= \frac{(n-1)n}{2} + \frac{2n}{2} \\ &= \frac{n^2 - n + 2n}{2} \\ &= \frac{n^2 + n}{2} \\ &= \frac{n(n+1)}{2} \end{aligned}$$

# The Sum $1 + 3 + \cdots + (2n - 1)$

## The recursive formula

$$S(n) = \begin{cases} 1 & \text{for } n = 1 \\ S(n-1) + (2n-1) & \text{for } n > 1 \end{cases}$$

## The recursive pseudocode

**function** $S(n)$ (* integer $n \geq 1$ *)
  **if** $n = 1$
  **then return** $(1)$
  **else return** $(S(n-1) + (2n-1))$

## The non-recursive pseudocode

**function** $S(n)$ (* integer $n \geq 1$ *)
  $s = 1$
  **for** $i = 2$ **to** $n$
    $s = s + (2i - 1)$
  **return**$(s)$

## The Sum $1 + 3 + \cdots + (2n - 1)$

### Bottom-Up evaluation

- $S(1) = \quad = 1 = 1^2$
- $S(2) = S(1) + (2 \cdot 2 - 1) = 1 + 3 = 4 = 2^2$
- $S(3) = S(2) + (2 \cdot 3 - 1) = 4 + 5 = 9 = 3^2$
- $S(4) = S(3) + (2 \cdot 4 - 1) = 9 + 7 = 16 = 4^2$
- $S(5) = S(4) + (2 \cdot 5 - 1) = 16 + 9 = 25 = 5^2$
- $S(6) = S(5) + (2 \cdot 6 - 1) = 25 + 11 = 36 = 6^2$
- $S(7) = S(6) + (2 \cdot 7 - 1) = 36 + 13 = 49 = 7^2$

# The Sum $1 + 3 + \cdots + (2n-1)$

## The closed-form expression

$$S(n) = n^2$$

## Proof by induction

- Induction base: $S(1) = 1 = 1^2$
- Induction hypothesis: $S(n-1) = (n-1)^2$ for $n > 1$
- Inductive step for $n > 1$:

$$\begin{aligned} S(n) &= S(n-1) + (2n-1) \\ &= (n-1)^2 + (2n-1) \\ &= n^2 - 2n + 1 + 2n - 1 \\ &= n^2 \end{aligned}$$

# Generalized Geometric Progressions

## The recursive formula

For real numbers $g_0$ and $d$ and a positive real number $q \neq 1$

$$G(n) = \begin{cases} g_0 & \text{for } n = 0 \\ qG(n-1) + d & \text{for } n > 0 \end{cases}$$

## The closed-form expression

$$G(n) = g_0 q^n + d\frac{q^n - 1}{q - 1}$$

## Proof

- By induction on $n \geq 0$

## Remark

- $G(n)$ is an arithmetic progression when $q = 1$ and $a_0 = g_0$.
- However the solution is different: $G(n) = g_0 + dn$.

# Generalized Geometric Progressions

## Top-Down evaluation

$$
\begin{aligned}
G(n) &= \quad qG(n-1)+d & &= q^1 G(n-1) + d(1) \\
&= q^1(qG(n-2)+d)+d(1) & &= q^2 G(n-2) + d(q+1) \\
&= q^2(qG(n-3)+d)+d(q+1) & &= q^3 G(n-3) + d(q^2+q+1) \\
&= q^3(qG(n-4)+d)+d(q^2+q+1) & &= q^4 G(n-4) + d(q^3+q^2+q+1) \\
&\ \vdots \\
&= q^{n-1}(qG(n-n))+d)+d(q^{n-2}+q^{n-3}+\cdots+q^2+q+1) \\
&= q^n G(0) + d(q^{n-1}+q^{n-2}+\cdots+q^2+q+1) \\
&= g_0 q^n + d\sum_{i=0}^{n-1} q^i \\
&= g_0 q^n + d\frac{q^n-1}{q-1}
\end{aligned}
$$

# Generalized Geometric Progressions

## Bottom-Up evaluation

$$
\begin{aligned}
G(0) &= g_0 \\
G(1) &= qG(0) + d = g_0 q + d \\
G(2) &= qG(1) + d = g_0 q^2 + dq + d \\
G(3) &= qG(2) + d = g_0 q^3 + dq^2 + dq + d \\
G(4) &= qG(3) + d = g_0 q^4 + dq^3 + dq^2 + dq + d \\
&\vdots \\
G(n) &= g_0 q^n + d \sum_{i=0}^{n-1} q^i \\
G(n) &= g_0 q^n + d \frac{q^n - 1}{q - 1}
\end{aligned}
$$

# Powers of $q \neq 1$

**The recursive formula for $g_0 = 1$ and $d = 0$**

$$G(n) = \begin{cases} 1 & \text{for } n = 0 \\ qG(n-1) & \text{for } n \geq 1 \end{cases}$$

**The closed-form expression**

$$\begin{aligned} G(n) &= g_0 \cdot q^n + d \cdot \frac{q^n - 1}{q - 1} \\ &= 1 \cdot q^n + 0 \cdot \frac{q^n - 1}{q - 1} \\ &= q^n \end{aligned}$$

# Sum of Powers of 2

### Definition

$$G(n) = \begin{cases} 0 & \text{for } n = 0 \\ 1 + 2 + \cdots + 2^{n-1} & \text{for } n \geq 1 \end{cases}$$

### Proposition

$$G(n) = \begin{cases} 0 & \text{for } n = 0 \\ 2G(n-1) + 1 & \text{for } n \geq 1 \end{cases}$$

### Proof

$$\begin{aligned} G(n) &= 1 + 2 + 4 + \cdots + 2^{n-1} \\ &= (2 + 4 + \cdots + 2^{n-1}) + 1 \\ &= 2(1 + 2 + \cdots + 2^{n-2}) + 1 \\ &= 2G(n-1) + 1 \end{aligned}$$

# Sum of Powers of 2

**The recursive formula for $g_0 = 0$, $d = 1$, and $q = 2$**

$$G(n) = \begin{cases} 0 & \text{for } n = 0 \\ 2G(n-1) + 1 & \text{for } n \geq 1 \end{cases}$$

**The closed-form expression**

$$\begin{aligned} G(n) &= g_0 \cdot q^n + d \cdot \frac{q^n - 1}{q - 1} \\ &= 0 \cdot 2^n + 1 \cdot \frac{2^n - 1}{2 - 1} \\ &= 2^n - 1 \end{aligned}$$

**Corollary**

$$1 + 2 + \cdots + 2^{n-1} = 2^n - 1$$

# Sum of Powers of $1/2$

## Definition

$$G(n) = \begin{cases} 0 & \text{for } n = 0 \\ 1/2 + 1/4 + \cdots + (1/2)^n & \text{for } n \geq 1 \end{cases}$$

## Proposition

$$G(n) = \begin{cases} 0 & \text{for } n = 0 \\ (1/2)G(n-1) + 1/2 & \text{for } n \geq 1 \end{cases}$$

## Proof

$$\begin{aligned} G(n) &= 1/2 + 1/4 + 1/8 + \cdots + (1/2)^n \\ &= (1/4 + 1/8 + \cdots + (1/2)^n) + 1/2 \\ &= (1/2)(1/2 + 1/4 + \cdots + (1/2)^{n-1}) + 1/2 \\ &= (1/2)G(n-1) + 1/2 \end{aligned}$$

# Sum of Powers of $1/2$

## The recursive formula for $g_0 = 0$, $d = 1/2$, and $q = 1/2$

$$G(n) = \begin{cases} 0 & \text{for } n = 0 \\ (1/2)G(n-1) + 1/2 & \text{for } n \geq 1 \end{cases}$$

## The closed-form expression

$$\begin{aligned} G(n) &= g_0 \cdot q^n + d \cdot \frac{q^n - 1}{q - 1} \\ &= 0 \cdot (1/2)^n + (1/2) \cdot \frac{(1/2)^n - 1}{(1/2) - 1} \\ &= (1/2) \cdot \frac{1 - (1/2)^n}{1 - 1/2} \\ &= 1 - (1/2)^n \end{aligned}$$

## Corollary

$$1/2 + 1/4 + 1/8 + \cdots + (1/2)^{n-1} = 1 - (1/2)^n$$

# Tower of Hanoi

## Definition by example

- https://www.youtube.com/watch?v=5Wn4EboLrMM

## Problem definition

- There are three pegs (rods) called $A$, $B$, and $C$ and $n \geq 1$ disks of different sizes.
- Initially all the disks are placed on peg $A$ ordered from the largest at the bottom to the smallest at the top.
- A **legal move** takes any top disk and moves it to another peg as long as it is not placed on top of a smaller disk.
- **Goal:** Move the $n$ disks from $A$ to $B$ using only legal moves.
- **Efficiency:** Move the disks with as few as possible legal moves.

## History, Background, and beyond

- https://en.wikipedia.org/wiki/Tower_of_Hanoi

# Tower of Hanoi

## Solution: demo

- https://www.mathsisfun.com/games/towerofhanoi.html

## General Recursive solution

- **Initial call:** Move $n \geq 1$ disks from $A$ to $B$
- **Recursive base:** For $n = 1$ move the single disk from $A$ to $B$
- **Recursive step:** Assume $k > 1$ disks are to be moved from peg $X$ to peg $Y$ for $X \neq Y$ and $\{X, Y, Z\} = \{A, B, C\}$:
  - Move the top $k - 1$ disks from $X$ to $Z$
  - Move the top disk from $X$ to $Y$
  - Move the top $k - 1$ disks from $Z$ to $Y$

## Recursive solution for four disks

- https://www.youtube.com/watch?v=YstLjLCGmgg

# Tower of Hanoi

## Correctness: proof by induction (sketch)

- **Induction base:** When $n = 1$, a largest top disk can be legally moved from a peg to an empty peg.
- **Induction hypothesis:** The smallest $1 \leq k < n$ disks can be legally moved from a peg to any of the other two pegs.
- **Inductive step:**
  - The $n - 1$ smallest disks are legally moved from peg $A$ to peg $C$ by the induction hypothesis.
  - The largest disk is legally moved from peg $A$ to the empty peg $B$.
  - The $n - 1$ smallest disks are legally moved from peg $C$ to peg $B$ which has the largest disk by the induction hypothesis.

## Total Number of moves

- Let $M(n)$ be the number of legal moves made by the recursive solution for $n \geq 1$ disks.
- Trivially, $M(1) = 1$ and by definition $M(0) = 0$.
- Recursively,

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

- The generalized geometric progression closed-form implies that

$$M(n) = 2^n - 1$$

# Tower of Hanoi

## Number of moves by disks

- For $1 \le i \le n$, let $m_k(n)$ be the number of legal moves of the $k^{\text{th}}$ disk made by the recursive solution for $n \ge 1$ disks.
- **Proposition:** $m_k(n) = 2^{k-1}$
- **Corollary:**

$$M(n) = \sum_{k=1}^{k=n} m_k(n) = \sum_{k=1}^{k=n} 2^{k-1} = \sum_{i=0}^{i=n-1} 2^i = 2^n - 1$$

# Fibonacci Numbers

**The sequence:**

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

**The recursive definition**

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

# Fibonacci Numbers

### The recursive pseudocode

**function** $F(n)$ (* integer $n \geq 0$ *)
  **if** $n = 0$ **then return** $(0)$
  **if** $n = 1$ **then return** $(1)$
  **otherwise return**$(F(n-1) + F(n-2))$

### The non-recursive pseudocode

**function** $F(n)$ (* integer $n \geq 0$ *)
  $F_0 = 0$
  $F_1 = 1$
  **for** $i = 2$ **to** $n$
    $F_i = F_{i-1} + F_{i-2}$
  **return**$(F_n)$

# Fibonacci Numbers - The Original Problem

## Story

- A just born pair of rabbits (one of each sex) is placed on an island.
- A pair of rabbits does not breed until they are 2 months old.
- After they are two months old, each pair of rabbits produces another pair each month.
- No rabbits ever die and no rabbits ever leave the island.

## Problem

- How many pairs of rabbits are there on the island after $n$ months?

## There are $F_n$ pairs of rabbits on the island after $n$ months

- The $F_{n-1}$ pairs of rabbits that were alive after $n-1$ months stay alive after $n$ months.
- The $F_{n-2}$ pairs of rabbits that were alive after $n-2$ months each produces a new pair of rabbits.

# Online Resources

## The Fibonacci's soup

- https://img.devrant.com/devrant/rant/r_2238362_6BfVK.jpg

## Encoding the Fibonacci Sequence into music

- https://www.youtube.com/watch?v=IGJeGOw8TzQ

## The original story

- https://www.youtube.com/watch?v=sjQlW6cH3Ko

## Some basics

- https://www.youtube.com/watch?v=ZC-d4dKTyKw

## Domino tilings of the $(2 \times n)$ – grid

- https://www.youtube.com/watch?v=AFAcKDTmYXI

# Additional Online Resources

## Texts

- Math is Fun:
  https://www.mathsisfun.com/numbers/fibonacci-sequence.html
- The life and numbers of Fibonacci:
  https://plus.maths.org/content/life-and-numbers-fibonacci
- Wikipedia:
  https://en.wikipedia.org/wiki/Fibonacci_number
- Fibonacci Numbers and the Golden Section:
  http://www.maths.surrey.ac.uk/hosted-sites/R.Knott/Fibonacci/fib.html

## Videos

- The magic of Fibonacci numbers:
  https://www.youtube.com/watch?v=SjSHVDfXHQ4&vl=ja
- The Fibonacci Sequence and Experies with Learning:
  https://www.youtube.com/watch?v=uk6CLffEuZM

# Three Consecutive Fibonacci Numbers

**Identity for $n \geq 1$**

$$F_{n-1}F_{n+1} = F_n^2 + (-1)^n$$

**Correctness for small $n$**

$$\begin{array}{rclclclcl}
F_0F_2 & = & 0 \cdot 1 & = & 0 & = & 1^2 - 1 & = & F_1^2 + (-1)^1 \\
F_1F_3 & = & 1 \cdot 2 & = & 2 & = & 1^2 + 1 & = & F_2^2 + (-1)^2 \\
F_2F_4 & = & 1 \cdot 3 & = & 3 & = & 2^2 - 1 & = & F_3^2 + (-1)^3 \\
F_3F_5 & = & 2 \cdot 5 & = & 10 & = & 3^2 + 1 & = & F_4^2 + (-1)^4 \\
F_4F_6 & = & 3 \cdot 8 & = & 24 & = & 5^2 - 1 & = & F_5^2 + (-1)^5 \\
F_5F_7 & = & 5 \cdot 13 & = & 65 & = & 8^2 + 1 & = & F_6^2 + (-1)^6 \\
F_6F_8 & = & 8 \cdot 21 & = & 168 & = & 13^2 - 1 & = & F_7^2 + (-1)^7
\end{array}$$

**“Almost” like integers and powers of 2**

$$\begin{array}{rcl}
(n-1)(n+1) & = & n^2 - 1 \\
2^{n-1} \cdot 2^{n+1} & = & (2^n)^2
\end{array}$$

# Proof By Induction

## Notations

$$\begin{aligned} L(n) &= F_{n-1}F_{n+1} \\ R(n) &= F_n^2 + (-1)^n \end{aligned}$$

## The induction base: $n = 1$

$$L(1) = F_0F_2 = 0 \cdot 1 = 0 = 1 - 1 = F_1^2 + (-1)^1 = R(1)$$

## The induction hypothesis: $L(n-1) = R(n-1)$ for $n > 1$

$$F_{n-2}F_n = F_{n-1}^2 + (-1)^{n-1}$$

$$F_{n-1}^2 = F_{n-2}F_n - (-1)^{n-1} = F_{n-2}F_n + (-1)^n$$

# Proof By Induction

**The inductive step:** $L(n) = R(n)$ **for** $n > 1$

$$
\begin{aligned}
L(n) &= F_{n-1}F_{n+1} \\
&= F_{n-1}(F_{n-1} + F_n) \\
&= F_{n-1}^2 + F_{n-1}F_n \\
&= F_{n-2}F_n + (-1)^n + F_{n-1}F_n \qquad (* \text{ the induction hypothesis } *) \\
&= F_{n-2}F_n + F_{n-1}F_n + (-1)^n \\
&= (F_{n-2} + F_{n-1})F_n + (-1)^n \\
&= F_n^2 + (-1)^n \\
&= R(n)
\end{aligned}
$$

# Sum of First $n$ Fibonacci numbers

**Identity for $n \geq 1$**

$$\sum_{i=1}^{n} F_i = F_1 + F_2 + \cdots + F_n = F_{n+2} - 1$$

**Correctness for small $n$**

$$\begin{aligned}
F_1 &= 1 &=& \quad 1 &=& \quad 2 - 1 = F_3 - 1 \\
F_1 + F_2 &= 1 + 1 &=& \quad 2 &=& \quad 3 - 1 = F_4 - 1 \\
F_1 + F_2 + F_3 &= 1 + 1 + 2 &=& \quad 4 &=& \quad 5 - 1 = F_5 - 1 \\
F_1 + F_2 + F_3 + F_4 &= 1 + 1 + 2 + 3 &=& \quad 7 &=& \quad 8 - 1 = F_6 - 1 \\
F_1 + F_2 + F_3 + F_4 + F_5 &= 1 + 1 + 2 + 3 + 5 &=& \quad 12 &=& \quad 13 - 1 = F_7 - 1
\end{aligned}$$

**"Almost" like powers of 2**

$$\sum_{i=0}^{n} 2^i = 1 + 2 + 4 + \cdots + 2^n = 2^{n+1} - 1$$

# Proof By Induction

## Notations

$$\begin{aligned} L(n) &= F_1 + F_2 + \cdots + F_n \\ R(n) &= F_{n+2} - 1 \end{aligned}$$

## The induction base: $n = 1$

$$L(1) = F_1 = 1 = 2 - 1 = F_3 - 1 = R(1)$$

## The induction hypothesis: $L(n-1) = R(n-1)$ for $n > 1$

$$F_1 + F_2 + \cdots + F_{n-1} = F_{n+1} - 1$$

# Proof By Induction

**The inductive step:** $L(n) = R(n)$ **for** $n > 1$

$$
\begin{aligned}
L(n) &= 1 + 1 + 2 + \cdots + F_{n-1} + F_n \\
&= L(n-1) + F_n \\
&= R(n-1) + F_n \qquad (* \text{ the induction hypothesis } *) \\
&= (F_{n+1} - 1) + F_n \\
&= (F_n + F_{n+1}) - 1 \\
&= F_{n+2} - 1 \\
&= R(n)
\end{aligned}
$$

# A Generalized Fibonacci Sequence

**Definition**

$$G_n = \begin{cases} a & \text{for } n = 0 \\ b & \text{for } n = 1 \\ G_{n-1} + G_{n-2} & \text{for } n \geq 2 \end{cases}$$

**Identity**

$$\sum_{i=0}^{n} G_i = G_0 + G_1 + \cdots + G_n = G_{n+2} - G_1 = G_{n+2} - b$$

**The Fibonacci sequence special case**

- $\sum_{i=0}^{n} G_i = G_{n+2} - 1$ for $a = 0$ and $b = 1$

**Example:** $a = 3$ **and** $b = 2$

- The sequence: $3, 2, 5, 7, 12, 19, 31, 50, 81, \ldots$
- The identity: $3 + 2 + 5 + 7 + 12 + 19 + 31 = 79 = 81 - 2$

# Proof Sketch

**Expanding $G_{n+2} - G_1$**

$$\begin{aligned}
G_{n+2} - G_1 &= G_n + G_{n+1} - G_1 \\
&= G_n + G_{n-1} + G_n - G_1 \\
&= G_n + G_{n-1} + G_{n-2} + G_{n-1} - G_1 \\
&= G_n + G_{n-1} + G_{n-2} + G_{n-3} + G_{n-2} - G_1 \\
&\vdots \\
&= G_n + G_{n-1} + G_{n-2} + \cdots + G_i + G_{i-1} + G_i - G_1 \\
&\vdots \\
&= G_n + G_{n-1} + G_{n-2} + \cdots + G_1 + G_0 + G_1 - G_1 \\
&= G_n + G_{n-1} + G_{n-2} + \cdots + G_1 + G_0
\end{aligned}$$

# Sum of First $n$ Odd-Indexed Fibonacci numbers

**Identity for $n \geq 1$**

$$\sum_{i=1}^{n} F_{2i-1} = F_1 + F_3 + \cdots + F_{2n-1} = F_{2n}$$

**Correctness for small $n$**

$$\begin{array}{lclcrcl}
F_1 & = & 1 & = & 1 & = & F_2 \\
F_1 + F_3 & = & 1+2 & = & 3 & = & F_4 \\
F_1 + F_3 + F_5 & = & 1+2+5 & = & 8 & = & F_6 \\
F_1 + F_3 + F_5 + F_7 & = & 1+2+5+13 & = & 21 & = & F_8 \\
F_1 + F_3 + F_5 + F_7 + F_9 & = & 1+2+5+13+34 & = & 55 & = & F_{10}
\end{array}$$

**"Almost" like powers of 2**

$$\sum_{i=0}^{n} 2^{2i-1} = 2 + 8 + 32 + \cdots + 2^{2n-1} = (2/3)(2^{2n} - 1)$$

# Proof I Sketch

## Expanding $F_{2n}$

$$
\begin{aligned}
F_{2n} &= F_{2n-1} + F_{2n-2} \\
&= F_{2n-1} + F_{2n-3} + F_{2n-4} \\
&= F_{2n-1} + F_{2n-3} + F_{2n-5} + F_{2n-6} \\
&= F_{2n-1} + F_{2n-3} + F_{2n-5} + F_{2n-7} + F_{2n-8} \\
&\vdots \\
&= F_{2n-1} + F_{2n-3} + F_{2n-5} + \cdots + F_{2k+1} + F_{2k} \\
&\vdots \\
&= F_{2n-1} + F_{2n-3} + F_{2n-5} + \cdots + F_3 + F_2 \\
&= F_{2n-1} + F_{2n-3} + F_{2n-5} + \cdots + F_3 + F_1 + F_0 \\
&= F_{2n-1} + F_{2n-3} + F_{2n-5} + \cdots + F_3 + F_1
\end{aligned}
$$

# Proof II Sketch

## Evaluating the sum

$$
\begin{aligned}
\sum_{i=1}^{n} F_{2i-1} &= F_1 + F_3 + F_5 + \cdots + F_{2n-3} + F_{2n-1} \\
&= (F_2 - F_0) + (F_4 - F_2) + (F_6 - F_4) + \cdots \\
&\qquad + (F_{2n-2} - F_{2n-4}) + (F_{2n} - F_{2n-2}) \\
&= F_{2n} - F_0 \\
&= F_{2n}
\end{aligned}
$$

# Sum of First $n$ Squares of Fibonacci numbers

## Identity for $n \geq 1$

$$\sum_{i=1}^{n} F_i^2 = F_1^2 + F_2^2 + \cdots + F_n^2 = F_n F_{n+1}$$

## Correctness for small $n$

$$\begin{aligned}
F_1^2 &= 1^2 &= 1 &= 1 \cdot 1 = F_1 F_2 \\
F_1^2 + F_2^2 &= 1^2 + 1^2 &= 2 &= 1 \cdot 2 = F_2 F_3 \\
F_1^2 + F_2^2 + F_3^2 &= 1^2 + 1^2 + 2^2 &= 6 &= 2 \cdot 3 = F_3 F_4 \\
F_1^2 + F_2^2 + F_3^2 + F_4^2 &= 1^2 + 1^2 + 2^2 + 3^2 &= 15 &= 3 \cdot 5 = F_4 F_5 \\
F_1^2 + F_2^2 + F_3^2 + F_4^2 + F_5^2 &= 1^2 + 1^2 + 2^2 + 3^2 + 5^2 &= 40 &= 5 \cdot 8 = F_5 F_6
\end{aligned}$$

## "Almost" like powers of 2

$$\sum_{i=0}^{n} (2^i)^2 = \sum_{i=0}^{n} 4^i = \frac{4^{n+1} - 1}{3} = \frac{2}{3} 2^n 2^{n+1} - \frac{1}{3}$$

# Proof By Induction

## Notations

$$\begin{aligned} L(n) &= F_1^2 + F_2^2 + \cdots + F_n^2 \\ R(n) &= F_n F_{n+1} \end{aligned}$$

## The induction base: $n = 1$

$$L(1) = F_1^2 = 1^2 = 1 = 1 \cdot 1 = F_1 F_2 = R(1)$$

## The induction hypothesis: $L(n-1) = R(n-1)$ for $n > 1$

$$F_1^2 + F_2^2 + \cdots + F_{n-1}^2 = F_{n-1} F_n$$

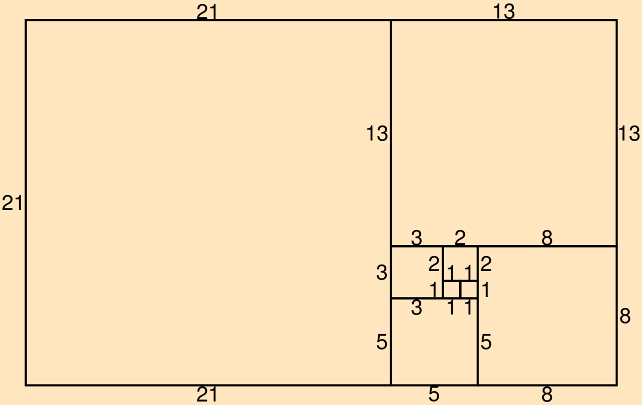
# Proof By Induction

**The inductive step:** $L(n) = R(n)$ **for** $n > 1$

$$\begin{aligned}
L(n) &= F_1^2 + F_2^2 + \cdots + F_{n-1}^2 + F_n^2 \\
&= L(n-1) + F_n^2 \\
&= R(n-1) + F_n^2 \qquad (* \text{ the induction hypothesis } *) \\
&= F_{n-1}F_n + F_n^2 \\
&= F_n(F_{n-1} + F_n) \\
&= F_nF_{n+1} \\
&= R(n)
\end{aligned}$$

# Proof Without Words: $\sum_{i=1}^{n} F_i^2 = F_n F_{n+1}$

## A figure




## A visual proof

- https://www.youtube.com/watch?v=2EsoiBdnIAM

# The Golden Ratio

## Definition

- The golden ratio $\phi$ is the positive root of the equation $x^2 = 1 + x$

## Formula

$$\phi = \frac{1+\sqrt{5}}{2} = 1.618\ldots$$

## The fractional part of the Golden Ratio is its reciprocal

$$\phi^2 = 1 + \phi \Longrightarrow \phi = \frac{1}{\phi} + 1 \Longrightarrow \phi - 1 = \frac{1}{\phi} = 0.618\ldots$$

# Online Resources

## A visual computation

- https://www.youtube.com/watch?v=yeHDXdv5KH4

## The Mona Lisa

- https://www.youtube.com/watch?v=jxKYFBtdsqU

## General in 3 minutes

- https://www.youtube.com/watch?v=fmaVqkR0ZXg

## General in 6 minutes

- https://www.youtube.com/watch?v=6nSfJEDZ_WM

# Infinite Identities for the Golden Ratio

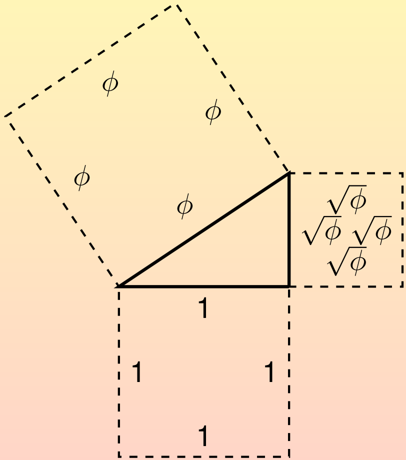
## Three identities

$$
\begin{aligned}
1 &= \frac{1}{\phi} + \frac{1}{\phi^3} + \frac{1}{\phi^5} + \frac{1}{\phi^7} \cdots \\
\phi &= 1 + \frac{1}{\phi^2} + \frac{1}{\phi^4} + \frac{1}{\phi^6} + \frac{1}{\phi^8} \cdots \\
\phi &= \frac{1}{\phi} + \frac{1}{\phi^2} + \frac{1}{\phi^3} + \frac{1}{\phi^4} + \frac{1}{\phi^5} \cdots
\end{aligned}
$$

## A visual proof

- https://www.youtube.com/watch?v=S181AOB2Bb8

# The Kepler Triangle




$$\phi^2 = 1^2 + (\sqrt{\phi})^2 = 1 + \phi$$

# The Golden ratio as a Function of Infinite 1's

## First infinite expressions for $\phi$

$$\phi = 1 + \cfrac{1}{1 + \cfrac{1}{1+\cfrac{1}{1+\cfrac{1}{1+\cdots}}}}$$

## Proof

- Define $F = 1 + \frac{1}{1+\frac{1}{1+\cdots}}$

$$F = 1 + \frac{1}{F} \implies F = \phi$$

# The Golden ratio as a Function of Infinite 1's

**Second infinite expressions for $\phi$**

$$\phi = \sqrt{1 + \sqrt{1 + \sqrt{1 + \cdots}}}$$

**Proof**

- Define $S = \sqrt{1 + \sqrt{1 + \sqrt{1 + \cdots}}}$

$$S = \sqrt{1 + S} \implies S^2 = 1 + S \implies S = \phi$$

# Solving the Equation $x = 1 + \frac{1}{x}$

**The iteration method:** $\phi = 1.6180339887498948204586834\ldots$

- Initially: $x_1 = 1$
- Iteratively: $x_i = 1 + \frac{1}{x_{i-1}}$ for $i > 1$
  - $x_2 = 1 + \frac{1}{1} = 2$
  - $x_3 = 1 + \frac{1}{2} = \frac{3}{2} = 1.5$
  - $x_4 = 1 + \frac{1}{3/2} = \frac{5}{3} \approx 1.667$
  - $x_5 = 1 + \frac{1}{5/3} = \frac{8}{5} = 1.6$
  - $x_6 = 1 + \frac{1}{8/5} = \frac{13}{8} = 1.625$
  - $x_7 = 1 + \frac{1}{13/8} = \frac{21}{13} \approx 1.61538461538$
  - $x_8 = 1 + \frac{1}{21/13} = \frac{34}{21} \approx 1.61904761905$
  - $x_9 = 1 + \frac{1}{34/21} = \frac{55}{34} \approx 1.61764705882$
  - $x_{20} = 1 + \frac{1}{6765/4181} = \frac{10946}{6765} \approx 1.61803399852$

# Solving the Equation $x = 1 + \frac{1}{x}$

$$x_1 = 1$$

$$x_2 = 1 + \frac{1}{1}$$

$$x_3 = 1 + \frac{1}{1 + \frac{1}{1}}$$

$$x_4 = 1 + \frac{1}{1 + \frac{1}{1+\frac{1}{1}}}$$

$$x_5 = 1 + \frac{1}{1 + \frac{1}{1+\frac{1}{1+\frac{1}{1}}}}$$

$$x = 1 + \frac{1}{1 + \frac{1}{1+\frac{1}{1+\frac{1}{1+\frac{1}{1+\frac{1}{1+\frac{1}{1+\frac{1}{1+\frac{1}{\cdots}}}}}}}}}$$

# Solving the Equation $x = 1 + \frac{1}{x}$

## Theorem

- $x_n = \frac{F_{n+1}}{F_n}$ for $n \geq 1$

## Proof By Induction

- Induction Base for $n = 1$: $x_1 = 1 = \frac{1}{1} = \frac{F_2}{F_1}$
- For $n > 1$, assume correctness for $x_{n-1}$ prove correctness for $x_n$

$$\begin{aligned}
x_n &= 1 + \frac{1}{x_{n-1}} \\
&= 1 + \frac{1}{F_n/F_{n-1}} && (\ast \text{ the induction hypothesis } \ast) \\
&= 1 + \frac{F_{n-1}}{F_n} \\
&= \frac{F_n + F_{n-1}}{F_n} \\
&= \frac{F_{n+1}}{F_n}
\end{aligned}$$

# Approximating $\phi$ with $\sqrt{1+\sqrt{1+\sqrt{1+\cdots}}}$

**The first 3 digits of the fractional part of $\phi = 1.6180339887\ldots$**

- Initially: $x_1 = \sqrt{1} = 1$
- Iteratively: $x_i = \sqrt{1 + x_{i-1}}$ for $i > 1$
  - $x_2 = \sqrt{1 + 1.000} = \sqrt{2.000} \approx 1.414$
  - $x_3 = \sqrt{1 + 1.414} = \sqrt{2.414} \approx 1.554$
  - $x_4 = \sqrt{1 + 1.554} = \sqrt{2.554} \approx 1.598$
  - $x_5 = \sqrt{1 + 1.598} = \sqrt{2.598} \approx 1.612$
  - $x_6 = \sqrt{1 + 1.612} = \sqrt{2.612} \approx 1.616$
  - $x_7 = \sqrt{1 + 1.616} = \sqrt{2.616} \approx 1.617$
  - $x_8 = \sqrt{1 + 1.617} = \sqrt{2.617} \approx 1.618$
  - $x_9 = \sqrt{1 + 1.618} = \sqrt{2.618} \approx 1.618$

# The Fibonacci Numbers and the Golden Ratio

## The two roots of the equation $x^2 - x - 1 = 0$

- The positive root: $\phi = \frac{1+\sqrt{5}}{2} \approx 1.618$
- The negative root: $\hat{\phi} = \frac{1-\sqrt{5}}{2} = 1 - \phi \approx -0.618$

## Fibonacci numbers as a function of the Golden Ratio

- $F_k = \frac{\phi^k - \hat{\phi}^k}{\sqrt{5}}$
- $F_{k+1} = \phi F_k + \hat{\phi}^k$
- $|\hat{\phi}| < 1 \implies F_k = \frac{\phi^k}{\sqrt{5}}$ rounded to the nearest integer
- $\frac{F_{k+1}}{F_k} \to \phi$ when $k \to \infty$