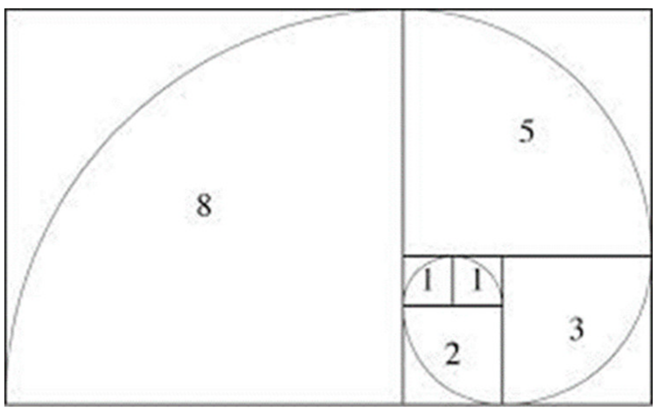
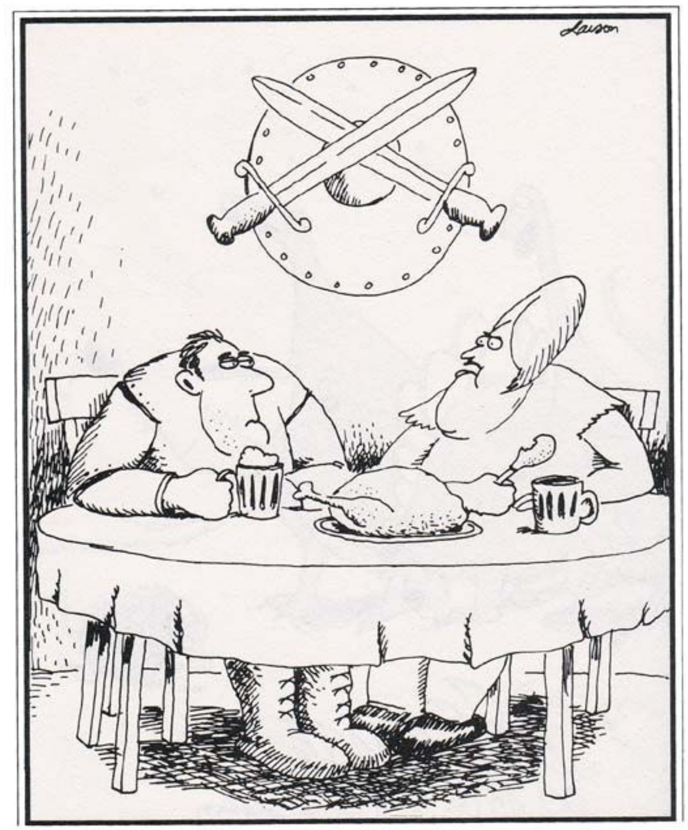
CSE 311: Foundations of Computing

Lecture 15: Recursion & Strong Induction Applications: Fibonacci & Euclid





"And another thing . . . I want you to be more assertive! I'm tired of everyone calling you Alexander the Pretty-Good!"

See Edstem post about 1-1 meetings with TAs not about current HW

Last class: Inductive Proofs In 5 Easy Steps

1. "Let $P(n)$ be... . We will show that $P(n)$ is true for all integers $n \geq b$ by induction."
2. "Base Case:" Prove $P(b)$
3. "Inductive Hypothesis:

   Assume that for some arbitrary integer $k \geq b$,

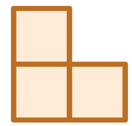
   $P(k)$ is true"
4. "Inductive Step:" Prove that $P(k+1)$ is true:

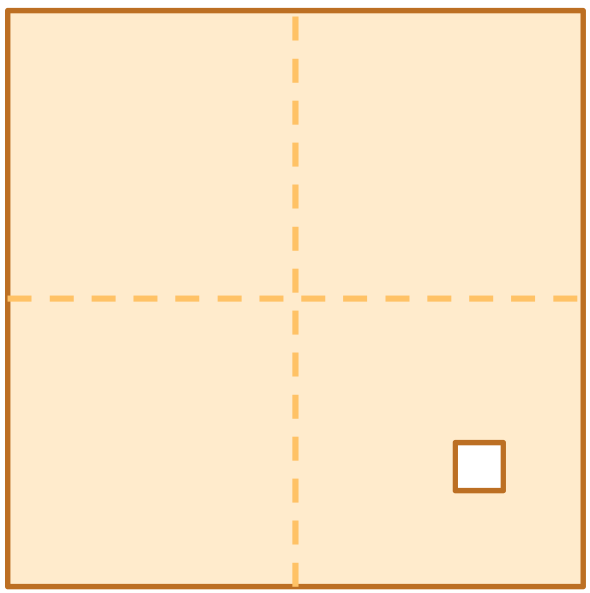
   Use the goal to figure out what you need.

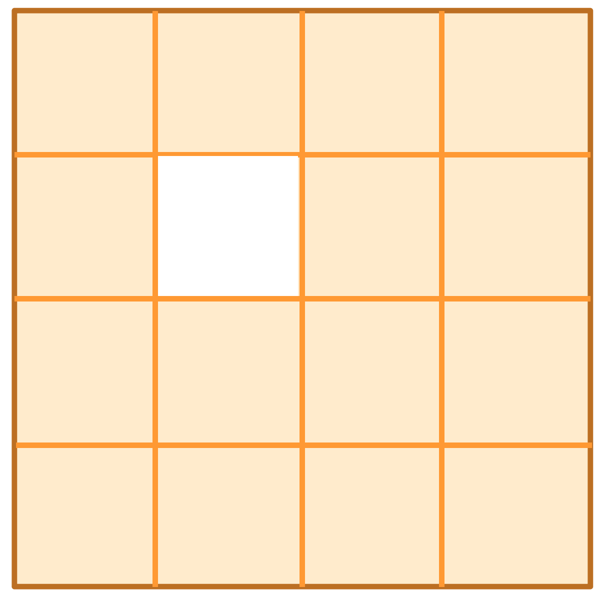
   Make sure you are using I.H. and point out where you are using it. (Don't assume $P(k+1)$!!)

5. "Conclusion: $P(n)$ is true for all integers $n \geq b$"

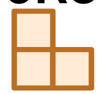
Checkerboard Tiling

- Prove that a $2^n \times 2^n$ checkerboard with one square removed can be tiled with: 

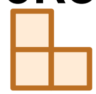




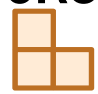
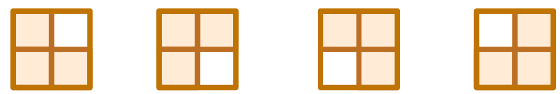
Checkerboard Tiling

1. **Let** P(n) **be "Any** $2^n \times 2^n$ **checkerboard with one square removed can be tiled with**  **" .**
   We prove P(n) **for all** $n \geq 1$ **by induction on** n.

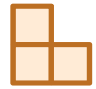
Checkerboard Tiling

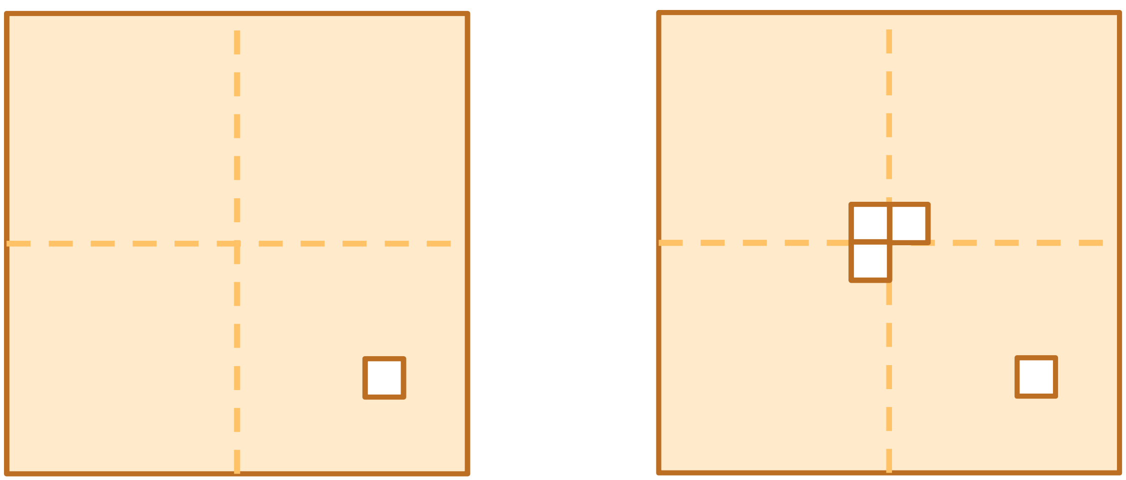
1. **Let** P(n) **be "Any** $2^n \times 2^n$ **checkerboard with one square removed can be tiled with**  **" .**
   We prove P(n) **for all** $n \geq 1$ **by induction on** n.
2. **Base Case:** n=1 

Checkerboard Tiling

1. **Let** $P(n)$ **be "Any** $2^n \times 2^n$ **checkerboard with one square removed can be tiled with**  **" .**
   We prove $P(n)$ **for all** $n \geq 1$ **by induction on** n.
2. **Base Case:** $n=1$ 
3. **Inductive Hypothesis: Assume** $P(k)$ **for some arbitrary integer** $k \geq 1$

Checkerboard Tiling

1. **Let** P(n) **be "Any** $2^n \times 2^n$ **checkerboard with one square removed can be tiled with**  **".**
   We prove P(n) **for all** $n \geq 1$ **by induction on** n.
2. **Base Case:** n=1 
3. **Inductive Hypothesis: Assume** P(k) **for some arbitrary integer** $k \geq 1$
4. **Inductive Step: Prove** P(k+1)



Apply IH to each quadrant then fill with extra tile.

Recall: Induction Rule of Inference

Domain: Natural Numbers

$$\begin{array}{c} P(0) \\ \forall k\ (P(k) \longrightarrow P(k+1)) \\ \hline \therefore \forall n\ \ P(n) \end{array}$$

How do the givens prove P(5)?

P(0)→P(1) P(1)→P(2) P(2)→P(3) P(3)→P(4) P(4)→P(5)

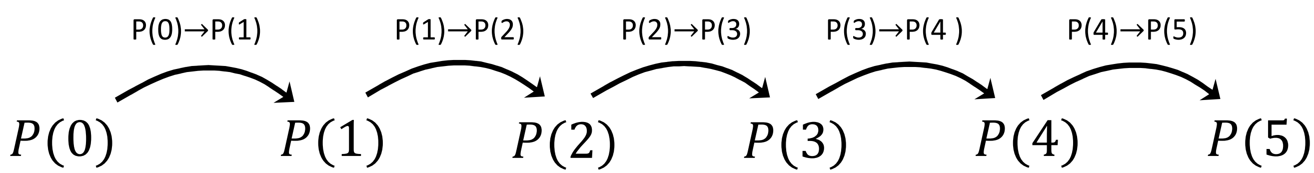
$$P(0) \quad P(1) \quad P(2) \quad P(3) \quad P(4) \quad P(5)$$

Recall: Induction Rule of Inference

Domain: Natural Numbers

$$\begin{array}{c} P(0) \\ \forall k\ (P(k) \longrightarrow P(k+1)) \\ \hline \therefore \forall n\ \ P(n) \end{array}$$

How do the givens prove P(5)?



We made it harder than we needed to …

- **When we proved** $P(2)$ **we knew BOTH** $P(0)$ **and** $P(1)$
- **When we proved** $P(3)$ **we knew** $P(0)$ **and** $P(1)$ **and** $P(2)$
- **When we proved** $P(4)$ **we knew** $P(0)$, $P(1)$, $P(2)$, $P(3)$
- **etc.**

That's the essence of the idea of Strong Induction.

Strong Induction

$$\frac{P(0) \qquad \forall k\ \Big(\forall j\ \big(0 \le j \le k \rightarrow P(j)\big) \rightarrow P(k+1)\Big)}{\therefore \forall n\ P(n)}$$

Strong Induction

$$\frac{P(0) \qquad \forall k\, \Big(\forall j\, \big(0 \le j \le k \to P(j)\big) \to P(k+1)\Big)}{\therefore \forall n\, P(n)}$$

Strong induction for P **follows from ordinary induction for** Q **where**

$$Q(k) := \forall j\, \big(0 \le j \le k \to P(j)\big)$$

Note that $Q(0) = P(0)$ **and** $Q(k+1) \equiv Q(k) \wedge P(k+1)$ **and** $\forall n\, Q(n) \equiv \forall n\, P(n)$

Last class: Inductive Proofs In 5 Easy Steps

1. "Let $P(n)$ be... . We will show that $P(n)$ is true for all integers $n \geq b$ by induction."
2. "Base Case:" Prove $P(b)$
3. "Inductive Hypothesis:

   Assume that for some arbitrary integer $k \geq b$,

   $P(k)$ is true"
4. "Inductive Step:" Prove that $P(k+1)$ is true:

   Use the goal to figure out what you need.

   Make sure you are using I.H. and point out where you are using it. (Don't assume $P(k+1)$!!)

5. "Conclusion: $P(n)$ is true for all integers $n \geq b$"

Strong Inductive Proofs In 5 Easy Steps

1. "Let $P(n)$ be... . We will show that $P(n)$ is true for all integers $n \geq b$ by *strong* induction."
2. "Base Case:" Prove $P(b)$
3. "Inductive Hypothesis:

   Assume that for some arbitrary integer $k \geq b$,

   $P(j)$ *is true for every integer* j *from* b to k"
4. "Inductive Step:" Prove that $P(k+1)$ is true:

   Use the goal to figure out what you need.

   Make sure you are using I.H. (that $P(b), \dots, P(k)$ *are true) and point out where you are using it.*
   (Don't assume $P(k+1)$ *!!)*
5. "Conclusion: $P(n)$ is true for all integers $n \geq b$"

Recall: Fundamental Theorem of Arithmetic

Every integer > 1 has a unique prime factorization

48 = 2 • 2 • 2 • 2 • 3
591 = 3 • 197
45,523 = 45,523
321,950 = 2 • 5 • 5 • 47 • 137
1,234,567,890 = 2 • 3 • 3 • 5 • 3,607 • 3,803

We use strong induction to prove that a factorization into primes exists, but not that it is unique.

Every integer ≥ 2 is a product of (one or more) primes.

Every integer ≥ 2 is a product of (one or more) primes.

1. **Let** P(n) **be** "n is a product of some list of primes". **We will show that** P(n) **is true for all integers** $n \geq 2$ **by strong induction.**

Every integer ≥ 2 is a product of (one or more) primes.

1. **Let** P(n) **be** "n is a product of some list of primes". **We will show that** P(n) **is true for all integers** $n \geq 2$ **by strong induction.**
2. **Base Case** (n=2): 2 **is prime, so it is a product of (one) prime.**
   Therefore P(2) **is true.**

Every integer ≥ 2 is a product of (one or more) primes.

1. **Let** P(n) **be** "n is a product of some list of primes". **We will show that** P(n) **is true for all integers** n $\geq$ 2 **by strong induction.**
2. **Base Case** (n=2): 2 **is prime, so it is a product of (one) prime.**
   Therefore P(2) **is true.**
3. **Inductive Hyp: Suppose that for some arbitrary integer** k ≥ 2**,**
   P(j) **is true for every integer** j **between** 2 **and** k

Every integer ≥ 2 is a product of (one or more) primes.

1. **Let** P(n) **be** "n is a product of some list of primes". **We will show that** P(n) **is true for all integers** $n \geq 2$ **by strong induction.**
2. **Base Case** (n=2): 2 **is prime, so it is a product of (one) prime.**
   Therefore P(2) **is true.**
3. **Inductive Hyp: Suppose that for some arbitrary integer** $k \geq 2$,
   P(j) **is true for every integer** j **between** 2 **and** k
4. **Inductive Step:**

   Goal: Show P(k+1); **i.e.** k+1 **is a product of primes**

Every integer ≥ 2 is a product of (one or more) primes.

1. **Let** P(n) **be** "n is a product of some list of primes". **We will show that** P(n) **is true for all integers** $n \geq 2$ **by strong induction.**
2. **Base Case** (n=2): 2 **is prime, so it is a product of (one) prime.**
   Therefore P(2) **is true.**
3. **Inductive Hyp: Suppose that for some arbitrary integer** $k \geq 2$,
   P(j) **is true for every integer** j **between** 2 **and** k
4. **Inductive Step:**

   Goal: Show P(k+1); **i.e.** k+1 **is a product of primes**

   <u>**Case:** k+1 **is prime**</u>**: Then by definition** k+1 **is a product of primes**

Every integer ≥ 2 is a product of (one or more) primes.

1. **Let** P(n) **be** "n is a product of some list of primes". **We will show that** P(n) **is true for all integers** $n \geq 2$ **by strong induction.**
2. **Base Case** (n=2): 2 **is prime, so it is a product of (one) prime.**
   Therefore P(2) **is true.**
3. **Inductive Hyp: Suppose that for some arbitrary integer** $k \geq 2$**,**
   P(j) **is true for every integer** j **between** 2 **and** k
4. **Inductive Step:**

   Goal: Show P(k+1); **i.e.** k+1 **is a product of primes**

   <u>Case:</u> <u>k+1</u> **<u>is prime</u>: Then by definition** k+1 **is a product of primes**
   <u>Case:</u> <u>k+1</u> **<u>is composite:</u> Then** k+1=ab **for some integers** a **and** b
   where $2 \leq a, b \leq k$.

Every integer ≥ 2 is a product of (one or more) primes.

1. **Let** $P(n)$ **be** "n is a product of some list of primes". **We will show that** $P(n)$ **is true for all integers** $n \geq 2$ **by strong induction.**
2. **Base Case** (n=2): 2 **is prime, so it is a product of (one) prime. Therefore** $P(2)$ **is true.**
3. **Inductive Hyp: Suppose that for some arbitrary integer** $k \geq 2$, $P(j)$ **is true for every integer** j **between** 2 **and** k
4. **Inductive Step:**

Goal: Show $P(k+1)$; **i.e.** $k+1$ **is a product of primes**

<u>Case: k+1 is prime</u>: Then by definition $k+1$ **is a product of primes**

<u>Case: k+1 is composite:</u> Then $k+1=ab$ **for some integers** a **and** b **where** $2 \leq a, b \leq k$. **By our IH,** $P(a)$ **and** $P(b)$ **are true so we have**

$a = p_1 p_2 \cdots p_r$ **and** $b = q_1 q_2 \cdots q_s$

for some primes $p_1, p_2, \ldots, p_r, q_1, q_2, \ldots, q_s$.

Thus, $k+1 = ab = p_1 p_2 \cdots p_r q_1 q_2 \cdots q_s$ **which is a product of primes.**

Since $k \geq 2$, **one of these cases must happen and so** $P(k+1)$ **is true.**

Every integer ≥ 2 is a product of (one or more) primes.

1. **Let** P(n) **be** "n is a product of some list of primes". **We will show that** P(n) **is true for all integers** $n \geq 2$ **by strong induction.**
2. **Base Case** (n=2): 2 **is prime, so it is a product of (one) prime.** **Therefore** P(2) **is true.**
3. **Inductive Hyp: Suppose that for some arbitrary integer** $k \geq 2$, P(j) **is true for every integer** j **between** 2 **and** k
4. **Inductive Step:**

   Goal: Show P(k+1); **i.e.** k+1 **is a product of primes**

   Case: k+1 is prime: Then by definition k+1 **is a product of primes**

   Case: k+1 is composite: Then k+1=ab **for some integers** a **and** b **where** $2 \leq a, b \leq k$. **By our IH,** P(a) **and** P(b) **are true so we have**

   $a = p_1 p_2 \cdots p_r$ **and** $b = q_1 q_2 \cdots q_s$

   for some primes $p_1, p_2, \ldots, p_r, q_1, q_2, \ldots, q_s$.

   Thus, $k+1 = ab = p_1 p_2 \cdots p_r q_1 q_2 \cdots q_s$ **which is a product of primes.**

   Since $k \geq 2$, **one of these cases must happen and so** P(k+1) **is true.**
5. **Thus** P(n) **is true for all integers** $n \geq 2$, **by strong induction.**

Strong Induction is particularly useful when…

…we need to analyze methods that on input k make a recursive call for an input different from $k-1$.

e.g.: Recursive Modular Exponentiation:

- For exponent $k > 0$ it made a recursive call with exponent $\mathrm{j} = k/2$ when k was even or $\mathrm{j} = k-1$ when k was odd.

Fast Exponentiation

```
public static int FastModExp(int a, int k, int modulus) {

        if (k == 0) {
            return 1;

         } else if ((k % 2) == 0) {
            long temp = FastModExp(a,k/2,modulus);
            return (temp * temp) % modulus;

        } else {
            long temp = FastModExp(a,k-1,modulus);
            return (a * temp) % modulus;
        }

}
```

$$a^{2j} \bmod m = \left(a^j \bmod m\right)^2 \bmod m$$

$$a^{2j+1} \bmod m = \left((a \bmod m) \cdot (a^{2j} \bmod m)\right) \bmod m$$

Strong Induction is particularly useful when…

…we need to analyze methods that on input k make a recursive call for an input different from $k-1$.

e.g.: Recursive Modular Exponentiation:

- For exponent $k > 0$ it made a recursive call with exponent $\mathrm{j} = k/2$ when k was even or $\mathrm{j} = k-1$ when k was odd.

We won't analyze this particular method by strong induction, but we could.

However, we will use strong induction to analyze other functions with recursive definitions.

Recursive definitions of functions

- $0! = 1$; $(n+1)! = (n+1) \cdot n!$ **for all** $n \geq 0$.
- $F(0) = 0$; $F(n+1) = F(n) + 1$ **for all** $n \geq 0$.
- $G(0) = 1$; $G(n+1) = 2 \cdot G(n)$ **for all** $n \geq 0$.
- $H(0) = 1$; $H(n+1) = 2^{H(n)}$ **for all** $n \geq 0$.

Prove $n! \leq n^n$ for all $n \geq 1$

1. **Let** P(n) **be** "n! ≤ n^n". **We will show that** P(n) **is true for all integers** n ≥ 1 **by induction.**
2. **Base Case** (n=1): $1!=1\cdot 0!=1\cdot 1=1=1^1$ **so** P(1) **is true.**
3. **Inductive Hypothesis: Suppose that** P(k) **is true for some arbitrary integer** k ≥ 1. **I.e., suppose** $k! \leq k^k$.

Prove $n! \leq n^n$ for all $n \geq 1$

1. **Let** P(n) **be** "$n! \leq n^n$". **We will show that** P(n) **is true for all integers** $n \geq 1$ **by induction.**
2. **Base Case** (n=1): $1!=1\cdot 0!=1\cdot 1=1=1^1$ **so** P(1) **is true.**
3. **Inductive Hypothesis: Suppose that** P(k) **is true for some arbitrary integer** $k \geq 1$. **I.e., suppose** $k! \leq k^k$.
4. **Inductive Step:**

   Goal: Show P(k+1), **i.e. show** $(k+1)! \leq (k+1)^{k+1}$

   $(k+1)! = (k+1)\cdot k!$ **by definition of !**

   $\leq (k+1)\cdot k^k$ **by the IH**

   $\leq (k+1)\cdot (k+1)^k$ **since** $k \geq 0$

   $= (k+1)^{k+1}$

   Therefore P(k+1) **is true.**
5. **Thus** P(n) **is true for all** $n \geq 1$, **by induction.**

More Recursive Definitions

Suppose that $h: \mathbb{N} \to \mathbb{R}$.

Then we have familiar summation notation:

$\sum_{i=0}^{0} h(i) = h(0)$

$\sum_{i=0}^{n+1} h(i) = h(n+1) + \sum_{i=0}^{n} h(i)$ **for** $n \geq 0$

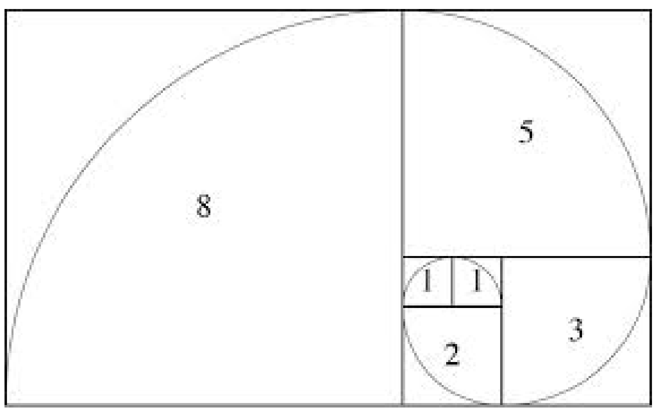
There is also product notation:

$\prod_{i=0}^{0} h(i) = h(0)$

$\prod_{i=0}^{n+1} h(i) = h(n+1) \cdot \prod_{i=0}^{n} h(i)$ **for** $n \geq 0$

Fibonacci Numbers

$$f_0 = 0$$
$$f_1 = 1$$
$$f_n = f_{n-1} + f_{n-2}$$ **for all** $n \geq 2$

Fibonacci Numbers

$f_0 = 0$
$f_1 = 1$
$f_n = f_{n-1} + f_{n-2}$ **for all** $n \geq 2$

Tamás Görbe
@TamasGorbe

A Mathematician's Way* of Converting Miles to Kilometers

$$\begin{aligned} 3 \text{ mi} &\approx 5 \text{ km} \\ 5 \text{ mi} &\approx 8 \text{ km} \\ 8 \text{ mi} &\approx 13 \text{ km} \end{aligned}$$

f_n mi ≈ f_{n+1} km

Bounding Fibonacci I: $f_n < 2^n$ for all $n \geq 0$

1. **Let** P(n) **be** "$f_n < 2^n$". **We prove that** P(n) **is true for all integers** $n \geq 0$ **by strong induction.**

$$f_0 = 0 \qquad f_1 = 1$$
$$f_n = f_{n-1} + f_{n-2} \text{ for all } n \geq 2$$

Bounding Fibonacci I: $f_n < 2^n$ for all $n \geq 0$

1. **Let** P(n) **be** "$f_n < 2^n$". **We prove that** P(n) **is true for all integers** $n \geq 0$ **by strong induction.**
2. **Base Case:** $f_0=0 < 1= 2^0$ **so** P(0) **is true.**

$f_0 = 0 \quad f_1 = 1$
$f_n = f_{n-1} + f_{n-2}$ for all $n \geq 2$

Bounding Fibonacci I: $f_n < 2^n$ for all $n \geq 0$

1. **Let** P(n) **be** "$f_n < 2^n$". **We prove that** P(n) **is true for all integers** $n \geq 0$ **by strong induction.**
2. **Base Case:** $f_0=0 < 1= 2^0$ **so** P(0) **is true.**
3. **Inductive Hypothesis: Assume that for some arbitrary integer** $k \geq 0$, **we have** $f_j < 2^j$ **for every integer** j **from** 0 **to** k.

$f_0 = 0 \quad f_1 = 1$
$f_n = f_{n-1} + f_{n-2}$ for all $n \geq 2$

Bounding Fibonacci I: $f_n < 2^n$ for all $n \geq 0$

1. **Let** P(n) **be "**$f_n < 2^n$**". We prove that** P(n) **is true for all integers** $n \geq 0$ **by strong induction.**
2. **Base Case:** $f_0 = 0 < 1 = 2^0$ **so** P(0) **is true.**
3. **Inductive Hypothesis: Assume that for some arbitrary integer** $k \geq 0$, **we have** $f_j < 2^j$ **for every integer** j **from** 0 **to** k.
4. **Inductive Step:** **Goal: Show** P(k+1); **that is,** $f_{k+1} < 2^{k+1}$

$f_0 = 0 \quad f_1 = 1$
$f_n = f_{n-1} + f_{n-2}$ for all $n \geq 2$

Bounding Fibonacci I: $f_n < 2^n$ for all $n \geq 0$

1. **Let** P(n) **be "**$f_n < 2^n$**". We prove that** P(n) **is true for all integers** $n \geq 0$ **by strong induction.**
2. **Base Case:** $f_0 = 0 < 1 = 2^0$ **so** P(0) **is true.**
3. **Inductive Hypothesis: Assume that for some arbitrary integer** $k \geq 0$**, we have** $f_j < 2^j$ **for every integer** j **from** 0 **to** k**.**
4. **Inductive Step:** **Goal: Show** P(k+1)**; that is,** $f_{k+1} < 2^{k+1}$

   Case k+1 = 1:

   Case k+1 ≥ 2:

$$f_0 = 0 \quad f_1 = 1$$
$$f_n = f_{n-1} + f_{n-2} \text{ for all } n \geq 2$$

Bounding Fibonacci I: $f_n < 2^n$ for all $n \geq 0$

1. **Let** P(n) **be "**$f_n < 2^n$**". We prove that** P(n) **is true for all integers** $n \geq 0$ **by strong induction.**
2. **Base Case:** $f_0=0 < 1= 2^0$ **so** P(0) **is true.**
3. **Inductive Hypothesis: Assume that for some arbitrary integer** $k \geq 0$, **we have** $f_j < 2^j$ **for every integer** j **from** 0 **to** k.
4. **Inductive Step:** **Goal: Show** P(k+1); **that is,** $f_{k+1} < 2^{k+1}$

   <u>Case k+1 = 1</u>: **Then** $f_1 = 1 < 2 = 2^1$ **so** P(k+1) **is true here.**

   <u>Case k+1 ≥ 2</u>:

$f_0 = 0 \quad f_1 = 1$
$f_n = f_{n-1} + f_{n-2}$ for all $n \geq 2$

Bounding Fibonacci I: $f_n < 2^n$ for all $n \geq 0$

1. Let P(n) be "$f_n < 2^n$". We prove that P(n) is true for all integers $n \geq 0$ by strong induction.
2. Base Case: $f_0=0 < 1= 2^0$ so P(0) is true.
3. Inductive Hypothesis: Assume that for some arbitrary integer $k \geq 0$, we have $f_j < 2^j$ for every integer j from 0 to k.
4. Inductive Step: Goal: Show P(k+1); that is, $f_{k+1} < 2^{k+1}$

   Case k+1 = 1: Then $f_1 = 1 < 2 = 2^1$ so P(k+1) is true here.

   Case $k+1 \geq 2$: Then $f_{k+1} = f_k + f_{k-1}$ by definition

   $< 2^k + 2^{k-1}$ by the IH since $k-1 \geq 0$

   $< 2^k + 2^k = 2 \cdot 2^k$

   $= 2^{k+1}$

   so P(k+1) is true in this case.

These are the only cases so P(k+1) follows.

$f_0 = 0 \quad f_1 = 1$

$f_n = f_{n-1} + f_{n-2}$ for all $n \geq 2$

Bounding Fibonacci I: $f_n < 2^n$ for all $n \geq 0$

1. **Let** P(n) **be "**$f_n < 2^n$**". We prove that** P(n) **is true for all integers** $n \geq 0$ **by strong induction.**
2. **Base Case:** $f_0 = 0 < 1 = 2^0$ **so** P(0) **is true.**
3. **Inductive Hypothesis: Assume that for some arbitrary integer** $k \geq 0$**, we have** $f_j < 2^j$ **for every integer** j **from** 0 **to** k.
4. **Inductive Step:** **Goal: Show** P(k+1)**; that is,** $f_{k+1} < 2^{k+1}$

   Case k+1 = 1: **Then** $f_1 = 1 < 2 = 2^1$ **so** P(k+1) **is true here.**

   Case $k+1 \geq 2$: **Then** $f_{k+1} = f_k + f_{k-1}$ **by definition**

   $< 2^k + 2^{k-1}$ **by the IH since** $k-1 \geq 0$

   $< 2^k + 2^k = 2 \cdot 2^k = 2^{k+1}$

   so P(k+1) **is true in this case.**

   These are the only cases so P(k+1) **follows.**
5. **Therefore by strong induction,**

   $f_n < 2^n$ **for all integers** $n \geq 0$.

$f_0 = 0 \quad f_1 = 1$

$f_n = f_{n-1} + f_{n-2}$ for all $n \geq 2$

Inductive Proofs with Multiple Base Cases

1. "Let $P(n)$ be... . We will show that $P(n)$ is true for all integers $n \geq b$ by induction."
2. "Base Cases:" Prove $P(b)$, $P(b+1)$, ..., $P(c)$
3. "Inductive Hypothesis:

   Assume $P(k)$ is true for an arbitrary integer $k \geq c$"
4. "Inductive Step:" Prove that $P(k+1)$ is true:

   Use the goal to figure out what you need.

   Make sure you are using I.H. and point out where you are using it. (Don't assume $P(k+1)$!!)
5. "Conclusion: $P(n)$ is true for all integers $n \geq b$"

Inductive Proofs With Multiple Base Cases

1. "Let $P(n)$ be... . We will show that $P(n)$ is true for all integers $n \geq b$ by ***strong*** induction."
2. "Base Cases:" Prove $P(b), P(b+1), \ldots, P(c)$
3. "Inductive Hypothesis:

   Assume that for some arbitrary integer $k \geq c$,

   $P(j)$ ***is true for every integer*** j ***from*** b **to** k"
4. "Inductive Step:" Prove that $P(k+1)$ is true:

   Use the goal to figure out what you need.

   Make sure you are using I.H. (that $P(b), \ldots, P(k)$ ***are true) and point out where you are using it.***
   (Don't assume $P(k+1)$ ***!!)***
5. "Conclusion: $P(n)$ is true for all integers $n \geq b$"

Original Version

Bounding Fibonacci I: $f_n < 2^n$ for all $n \geq 0$

1. **Let** P(n) **be "**$f_n < 2^n$**". We prove that** P(n) **is true for all integers** $n \geq 0$ **by strong induction.**
2. **Base Case:** $f_0 = 0 < 1 = 2^0$ **so** P(0) **is true.**
3. **Inductive Hypothesis: Assume that for some arbitrary integer** $k \geq 0$**, we have** $f_j < 2^j$ **for every integer** j **from** 0 **to** k**.**
4. **Inductive Step:** **Goal: Show** P(k+1)**; that is,** $f_{k+1} < 2^{k+1}$

   Case k+1 = 1: **Then** $f_1 = 1 < 2 = 2^1$ **so** P(k+1) **is true here.**

   Case k+1 ≥ 2: **Then** $f_{k+1} = f_k + f_{k-1}$ **by definition**

   $< 2^k + 2^{k-1}$ **by the IH since** $k-1 \geq 0$

   $< 2^k + 2^k = 2 \cdot 2^k = 2^{k+1}$

   so P(k+1) **is true in this case.**

   These are the only cases so P(k+1) **follows.**

   First case in inductive step didn't need IH

5. **Therefore by strong induction,**

   $f_n < 2^n$ **for all integers** $n \geq 0$.

$f_0 = 0 \quad f_1 = 1$

$f_n = f_{n-1} + f_{n-2}$ for all $n \geq 2$

Multiple Base Case Version

Bounding Fibonacci I: $f_n < 2^n$ for all $n \geq 0$

1. Let P(n) be "$f_n < 2^n$". We prove that P(n) is true for all integers $n \geq 0$ by strong induction.
2. Base Cases: $f_0 = 0 < 1 = 2^0$ so P(0) is true.
   $f_1 = 1 < 2 = 2^1$ so P(1) is true.
3. Inductive Hypothesis: Assume that for some arbitrary integer $k \geq 1$, we have $f_j < 2^j$ for every integer j from 0 to k.
4. Inductive Step: Goal: Show P(k+1); that is, $f_{k+1} < 2^{k+1}$

   We have $f_{k+1} = f_k + f_{k-1}$ by definition since $k+1 \geq 2$

   $< 2^k + 2^{k-1}$ by the IH since $k-1 \geq 0$

   $< 2^k + 2^k = 2 \cdot 2^k = 2^{k+1}$

   so P(k+1) is true.
5. Therefore, by strong induction, $f_n < 2^n$ for all integers $n \geq 0$.

Two base cases

Largest base case

Smallest base case

Two base cases, and two previous values used

$f_0 = 0 \quad f_1 = 1$

$f_n = f_{n-1} + f_{n-2}$ for all $n \geq 2$

Bounding Fibonacci II: $f_n \geq 2^{n/2-1}$ for all $n \geq 2$

1. Let P(n) be "$f_n \geq 2^{n/2-1}$". We prove that P(n) is true for all integers $n \geq 2$ by strong induction.

Two base cases, and two previous values used

$f_0 = 0 \quad f_1 = 1$
$f_n = f_{n-1} + f_{n-2}$ for all $n \geq 2$

Bounding Fibonacci II: $f_n \geq 2^{n/2 - 1}$ for all $n \geq 2$

1. **Let** P(n) **be "**$f_n \geq 2^{n/2 - 1}$**". We prove that** P(n) **is true for all integers** $n \geq 2$ **by strong induction.**
2. **Base Cases:** $f_2 = f_1 + f_0 = 1$ and $2^{2/2 - 1} = 2^0 = 1$ **so** P(2) **holds**
   $f_3 = f_2 + f_1 = 2 \geq 2^{1/2} = 2^{3/2-1}$ **so** P(3) **holds**

$f_0 = 0 \quad f_1 = 1$
$f_n = f_{n-1} + f_{n-2}$ for all $n \geq 2$

Bounding Fibonacci II: $f_n \geq 2^{n/2-1}$ for all $n \geq 2$

1. **Let** P(n) **be "**$f_n \geq 2^{n/2-1}$**". We prove that** P(n) **is true for all integers** $n \geq 2$ **by strong induction.**
2. **Base Cases:** $f_2 = f_1 + f_0 = 1$ and $2^{2/2-1} = 2^0 = 1$ **so** P(2) **holds**
   $f_3 = f_2 + f_1 = 2 \geq 2^{1/2} = 2^{3/2-1}$ **so** P(3) **holds**
3. **Inductive Hypothesis: Assume that for some arbitrary integer** $k \geq 3$, P(j) **is true for every integer** j **from** 2 **to** k.

$f_0 = 0 \quad f_1 = 1$

$f_n = f_{n-1} + f_{n-2}$ for all $n \geq 2$

Bounding Fibonacci II: $f_n \geq 2^{n/2-1}$ for all $n \geq 2$

1. **Let** P(n) **be "**$f_n \geq 2^{n/2-1}$**". We prove that** P(n) **is true for all integers** $n \geq 2$ **by strong induction.**
2. **Base Cases:** $f_2 = f_1 + f_0 = 1$ and $2^{2/2-1} = 2^0 = 1$ **so** P(2) **holds**
   $f_3 = f_2 + f_1 = 2 \geq 2^{1/2} = 2^{3/2-1}$ **so** P(3) **holds**
3. **Inductive Hypothesis: Assume that for some arbitrary integer** $k \geq 3$, P(j) **is true for every integer** j **from** 2 **to** k**.**
4. **Inductive Step:** **Goal: Show** P(k+1)**; that is,** $f_{k+1} \geq 2^{(k+1)/2-1}$

$f_0 = 0 \quad f_1 = 1$
$f_n = f_{n-1} + f_{n-2}$ for all $n \geq 2$

Bounding Fibonacci II: $f_n \geq 2^{n/2-1}$ for all $n \geq 2$

1. **Let** $P(n)$ **be "**$f_n \geq 2^{n/2-1}$**". We prove that** $P(n)$ **is true for all integers** $n \geq 2$ **by strong induction.**
2. **Base Cases:** $f_2 = f_1 + f_0 = 1$ and $2^{2/2-1} = 2^0 = 1$ **so** $P(2)$ **holds**
   $f_3 = f_2 + f_1 = 2 \geq 2^{1/2} = 2^{3/2-1}$ **so** $P(3)$ **holds**
3. **Inductive Hypothesis: Assume that for some arbitrary integer** $k \geq 3$, $P(j)$ **is true for every integer** j **from** 2 **to** k**.**
4. **Inductive Step:** **Goal: Show** $P(k+1)$**; that is,** $f_{k+1} \geq 2^{(k+1)/2-1}$

   We have $f_{k+1} = f_k + f_{k-1}$ **by definition since** $k+1 \geq 2$

   $\geq 2^{k/2-1} + 2^{(k-1)/2-1}$ **by the IH since** $k-1 \geq 2$

   $\geq 2^{(k-1)/2-1} + 2^{(k-1)/2-1} = 2^{(k-1)/2} = 2^{(k+1)/2-1}$

   so $P(k+1)$ **is true.**
5. **Therefore by strong induction,** $f_n \geq 2^{n/2-1}$ **for all integers** $n \geq 2$.

$f_0 = 0 \quad f_1 = 1$

$f_n = f_{n-1} + f_{n-2}$ for all $n \geq 2$

Running time of Euclid's algorithm

Theorem: Suppose that Euclid's Algorithm takes n steps for $\gcd(a, b)$ with $a \geq b > 0$. Then, $a \geq f_{n+1}$.

Running time of Euclid's algorithm

Theorem: Suppose that Euclid's Algorithm takes n steps for $\gcd(a, b)$ with $a \geq b > 0$. Then, $a \geq f_{n+1}$.

Why does this help us bound the running time of Euclid's Algorithm?

We already proved that $f_n \geq 2^{n/2 - 1}$ so $f_{n+1} \geq 2^{(n-1)/2}$

Therefore: if Euclid's Algorithm takes n steps
for $\gcd(a, b)$ with $a \geq b > 0$
then $a \geq 2^{(n-1)/2}$

so $(n-1)/2 \leq \log_2 a$ or $n \leq 1 + 2\log_2 a$
i.e., # of steps ≤ 1 + twice the # of bits in a.

Running time of Euclid's algorithm

Theorem: **Suppose that Euclid's Algorithm takes** n **steps for** $\gcd(a, b)$ **with** $a \geq b > 0$. **Then,** $a \geq f_{n+1}$.

An informal way to get the idea: Consider an n **step** gcd **calculation starting with** r_{n+1}=a **and** r_n=b:

$$
\begin{aligned}
r_{n+1} &= q_n r_n + r_{n-1} \\
r_n &= q_{n-1} r_{n-1} + r_{n-2} \\
&\ \ \dots \\
r_3 &= q_2 r_2 + r_1 \\
r_2 &= q_1 r_1 + 0
\end{aligned}
$$

f_0

For all $k \geq 2$, $r_{k-1} = r_{k+1} \bmod r_k$

"Euclid's algorithm is slowest on Fibonacci numbers and it takes only n steps for $\gcd(f_{n+1}, f_n)$"

Now $r_1 \geq 1$ **and each** q_k **must be** ≥ 1. **If we replace all the** q_K's **by** 1 **and replace** r_1 **by** 1 **, we can only reduce the** r_k's. **After that reduction,** $r_k = f_k$ **for every** k.

Running time of Euclid's algorithm

Theorem: **Suppose that Euclid's Algorithm takes n steps for $\gcd(a, b)$ with $a \geq b > 0$. Then, $a \geq f_{n+1}$.**

We go by strong induction on n.

Let P(n) **be** "gcd(a,b) **with** a ≥ b>0 **takes** n **steps** → $a \geq f_{n+1}$" **for all** n ≥ 1.

<u>Base Case</u>: n=1 **Suppose Euclid's Algorithm with** a ≥ b > 0 **takes** 1 **step.**

By assumption, $a \geq b \geq 1 = f_2$ **so** P(1) **holds.**

n=2 **Suppose Euclid's Algorithm with** a ≥ b > 0 **takes** 2 **steps.**

Then a = q b + r

b = q' r + 0 **for** r ≥ 1.

Since a ≥ b > 0, **we must have** q ≥ 1 **and** b ≥1 **so**

$a = qb + r \geq b + r \geq 1+1 = 2 = f_3$ **and** P(2) **holds**

<u>Induction Hypothesis</u>: **Suppose that for some integer** k ≥ 2, P(j) **is true for all integers** j **s.t.** 1 ≤ j ≤ k

Running time of Euclid's algorithm

<u>Induction Hypothesis</u>: Suppose that for some integer $k \geq 2$, $P(j)$ **is true for all integers** j **s.t.** $1 \leq j \leq k$

<u>Inductive Step</u>: **Goal: if** gcd(a,b) **with** $a \geq b>0$ **takes** $k+1$ **steps, then** $a \geq f_{k+2.}$

Since $k \geq 2$, **if** gcd(a,b) **with** $a \geq b>0$ **takes** $k+1 \geq 3$ **steps, the first** 3 **steps of Euclid's algorithm on** a **and** b **give us**

$a = q\,b + r$

$b = q'\,r + r'$

$r = q''\,r' + r''$

and there are k-2 **more steps after this. Note that this means that the** gcd(b, r) **takes** k **steps and** gcd(r, r') **takes** k-1 **steps.**

So since k, $k-1 \geq 1$, **by the IH we have** $b \geq f_{k+1}$ **and** $r \geq f_k$.

Running time of Euclid's algorithm

Induction Hypothesis: Suppose that for some integer $k \geq 2$, $P(j)$ is true for all integers j s.t. $1 \leq j \leq k$

Inductive Step: Goal: if $\gcd(a,b)$ with $a \geq b>0$ takes $k+1$ steps, then $a \geq f_{k+2}$.

Since $k \geq 2$, if $\gcd(a,b)$ with $a \geq b>0$ takes $k+1 \geq 3$ steps, the first 3 steps of Euclid's algorithm on a and b give us

$$a = q\, b + r$$
$$b = q'\, r + r'$$
$$r = q''\, r' + r''$$

and there are $k-2$ more steps after this. Note that this means that the $\gcd(b, r)$ takes k steps and $\gcd(r, r')$ takes $k-1$ steps.

So since $k, k-1 \geq 1$, by the IH we have $b \geq f_{k+1}$ and $r \geq f_k$.

Also, since $a \geq b$, we must have $q \geq 1$.

So $a = q\, b + r \geq b + r \geq f_{k+1} + f_k = f_{k+2}$ as required. ■