Today's learning goals

- Explain the steps in a proof by (strong) mathematical induction
- Use (strong) mathematical induction to prove
  - correctness of identities and inequalities
  - properties of algorithms
  - properties of geometric constructions
- Represent functions in multiple ways
- Use functions to define sequences: arithmetic progressions, geometric progressions
- Use and prove properties of recursively defined functions and recurrence relations (using induction)
- Evaluate which variant of induction is appropriate in a given situation, and apply it accordingly
- Find mistakes in purported proofs using induction and explain why they are wrong.
Recursive definitions

The set of bit strings \( \{0,1\}^* \) is defined recursively by

**Basis step:** \( \lambda \in \{0,1\}^* \) where \( \lambda \) is the empty string.

**Recursive step:** If \( w \in \{0,1\}^* \), then \( w0, w1 \in \{0,1\}^* \) and

How many strings are formed using the basis step and at most \( k \) applications of the recursive step (\( k \geq 1 \))?

A. 1  
B. \( k \)  
C. \( k^2+k+1 \)  
D. \( 2^k+1 \)  
E. None of the above
Recursive definitions

The set of **bit strings** \(\{0,1\}^*\) is defined recursively by

**Basis step:** \(\lambda \in \{0,1\}^*\) where \(\lambda\) is the empty string.

**Recursive step:** If \(w \in \{0,1\}^*\), then \(w0 \in \{0,1\}^*\) and \(w1 \in \{0,1\}^*\).

*Where does the notation \(\{0,1\}^*\) come from?*  (cf. HW 6 Q2)

Kleene star applied to set \(A\) gives a set, written \(A^*\), defined as:

**Basis step:** \(\lambda \in A^*\)

**Recursive step:** If \(u\) is in \(A^*\) and \(v\) is in \(A\), then \(uv \in A^*\)
Recursive definition: rooted trees

The set of **rooted trees** is defined recursively:

**Basis step**: A single vertex $r$ is a rooted tree.

**Recursive step**: Suppose that $T_1, \ldots, T_n$ are disjoint rooted trees with roots $r_1, \ldots, r_n$ (respectively). Then there is a rooted tree consisting of a (new) root together with edges connecting this root to each of the vertices $r_1, \ldots, r_n$. 

Rosen p. 351
Recursive definition: eb trees  

Rosen p. 352

Basis step:

Recursive step

1. *What can* $T_1 \ldots T_n$ *be?*

2. *What can* $T_1 \ldots T_n$ *be?*
Recursive definition: rooted trees

The set of rooted trees is defined recursively:

**Basis step**: A single vertex \( r \) is a rooted tree.

**Recursive step**: Suppose that \( T_1, \ldots, T_n \) are disjoint rooted trees with roots \( r_1, \ldots, r_n \) (respectively). Then there is a rooted tree consisting of a (new) root together with edges connecting this root to each of the vertices \( r_1, \ldots, r_n \).

How many rooted trees are formed using the basis step and at most \( k \) applications of the recursive step (\( k \geq 1 \))?

A. 1
B. \( k \)
C. \( k^2 \)
D. \( 2^k \)
E. None of the above
And now for something completely different …

(except not really)
Nim

Two players start with two (equal) piles of jellybeans in front of them. They take turns removing any positive # of jellybeans at a time from one of two piles in front of them. The player who removes the last jellybean wins the game.
Nim

Two players start with two (equal) piles of jellybeans in front of them. They take turns removing any positive # of jellybeans at a time from one of two piles in front of them. The player who removes the last jellybean wins the game.

Who has a strategy to guarantee a win?
Nim

The piles start out with equal #s. Two players take turns removing any positive # of jellybeans at a time from one of two piles in front of them. The player who removes the last jellybean wins the game.

Who wins? P2
Nim

Two players take turns removing any positive # of jellybeans at a time from one of two piles in front of them. The piles start out with equal #s. The player who removes the last jellybean wins the game.

\[ n=2 \]

Who wins? \[ P_2 \]
Nim

Two players take turns removing any positive # of jellybeans at a time from one of two piles in front of them. The piles start out with equal #s. The player who removes the last jellybean wins the game.

Idea: 2nd player takes the same amount 1st player took but from opposite pile. Game reduces to same setup but with fewer jellybeans.
Strong induction

To show that some statement $P(n)$ is true about all positive integers $n$,

1. Verify that $P(1)$ is true.
2. Let $k$ be an arbitrary positive integer. Show that

$$[P(1) \land P(2) \land \cdots \land P(k)] \rightarrow P(k + 1)$$

is true.

Rosen p. 334
Nim

Two players take turns removing any positive # of jellybeans at a time from one of two piles in front of them. The piles start out with equal #s. The player who removes the last jellybean wins the game.

Theorem: the second player can always guarantee a win.

Proof: By Strong Mathematical Induction, on # jellybeans in each pile.

1. Basis step WTS if piles each have 1, then 2\textsuperscript{nd} player can win.
2. Induction step Let k be a positive integer. As the Strong Induction Hypothesis Assume that 2\textsuperscript{nd} player can win whenever there are at least j jellybeans in each pile, for each j between 1 and k (inclusive). WTS 2\textsuperscript{nd} player has winning strategy when start with k+1 jellybeans in each pile.
How does this connect?

Rosen Sec 5.3

\[ n! \quad 0! = 1, \quad n! = n \ (n-1)! \quad \text{for } n > 0. \]

\[ 2^n \quad 2^0 = 1, \quad 2^n = 2 \ (2^{n-1}) \quad \text{for } n > 0. \]

2, -8, 32, -128, 512, …

\[ a_0 = 2, \quad a_n = -4 \ a_{n-1} \quad \text{for } n > 0. \]

Recursive definitions can refer to much earlier terms / values
Fibonacci numbers

\[ f_0 = 1, \quad f_1 = 1, \quad f_n = f_{n-1} + f_{n-2} \]

What are some sample values?

How quickly do these values grow?

\[ f_0 = 1 \]
\[ f_1 = 1 \]
\[ f_2 = f_1 + f_0 = 2 \]
\[ f_3 = f_2 + f_1 = 3 \]
Theorem: For each integer $n \geq 2$, $f_n \geq 1.5^{n-2}$

Proof: By Strong Mathematical Induction, on $n \geq 2$.

1. **Basis step** WTS $f_2 \geq 1.5^{2-2}$.

2. **Induction step** Let $k$ be an integer, $k \geq 2$. Assume as the strong induction hypothesis that the inequality is true about $f_j$ for each integer $j$, $2 \leq j \leq k$. WTS statement is true about $f_{k+1}$.
Fibonacci numbers

\[ f_0 = 1, \quad f_1 = 1, \quad f_n = f_{n-1} + f_{n-2} \]

1. Basis step WTS \( f_2 \geq 1.5^{2-2} \).

\[
\begin{align*}
\text{LHS} &= f_2 = 1 + 1 = 2. \\
\text{RHS} &= 1.5^{2-2} = 1.5^0 = 1.
\end{align*}
\]

Since \( 2 > 1 \), LHS > RHS so, in particular, LHS \( \geq \) RHS ☺
Fibonacci numbers

\[ f_0 = 1, \ f_1 = 1, \ f_n = f_{n-1} + f_{n-2} \]

**Induction step** Let \( k \) be an integer with \( k \geq 2 \).
Assume as the **strong induction hypothesis** that

\[ f_j \geq 1.5^{j-2} \]

for each integer \( j \) with \( 2 \leq j \leq k \).
**WTS** that \( f_{k+1} \geq 1.5^{(k+1)-2} \)

By definition of Fibonacci numbers, since \( k+1 > 1 \), \( f_{k+1} = f_k + f_{k-1} \).
Therefore, \( \text{LHS} = f_{k+1} = f_k + f_{k-1} \).

*Idea: apply strong induction hypothesis to \( k \) and \( k-1 \). Can we do it?*
Case 1: $k=2$ and WTS that $f_3 \geq 1.5^{(3)-2}$

Case 2: $k>2$ and WTS that $f_{k+1} \geq 1.5^{(k+1)-2}$ and strong IH applies to $k, k-1$ (because both $k, k-1$ are greater than or equal to 2 and less than $k+1$).

So let's prove each of these cases in turn:

Case 1: $k=2$ and WTS that $f_3 \geq 1.5^{(3)-2}$

By definition of Fibonacci numbers, LHS = $f_3=f_2+f_1=2+1=3$.

By algebra, RHS = $1.5^{3-2}=1.5$ Since $3>1.5$, LHS>RHS 😊
Fibonacci numbers

... 

Case 2: k>2 and WTS that $f_{k+1} \geq 1.5^{(k+1)-2}$ and strong IH applies to k, k-1 (because both k, k-1 are greater than or equal to 2 and less than k+1).

$\text{LHS} = f_{k+1} = f_k + f_{k-1} \geq 1.5^{k-2} + 1.5^{(k-1)-2} = 1.5^{k-3}(1.5+1) = 1.5^{k-3}(2.5) > 1.5^{k-3}(2.25) = 1.5^{k-3}1.5^2 = 1.5^{k-1} = 1.5^{(k+1)-2} = \text{RHS}.$

Def of Fibonacci numbers

Strong induction hypothesis
Fibonacci numbers

\[ f_0 = 1, \quad f_1 = 1, \quad f_n = f_{n-1} + f_{n-2} \]

Theorem: For each integer \( n \geq 2 \), \( f_n \geq 1.5^{n-2} \)

Proof: By Strong Mathematical Induction, on \( n \geq 2 \).

1. **Basis step** WTS \( f_2 \geq 1.5^{2-2} \) and \( f_3 \geq 1.5^{3-2} \)

2. **Induction step** Let \( k \) be an integer, \( k \geq 3 \). Assume as the **strong induction hypothesis** that the inequality is true about \( f_j \) and \( f_{j-1} \) for each integer \( j \), \( 2 \leq j \leq k \). WTS statement is true about \( f_{k+1} \).
Flavors of induction

- Mathematical induction
  Domain: $\{x \in \mathbb{Z} / x \geq b\}$

- Structural induction
  Domain: any recursive definition set

- Strong induction
  Domain: $\{x \in \mathbb{Z} / x \geq b\}$
Looking back

- We now have all the tools we need to rigorously prove
  - Correctness of **greedy change-making algorithm** with quarters, dimes, nickels, and pennies *Proof by contradiction, Rosen p. 199*
  - The **division algorithm** is correct *Strong induction, Rosen p. 341*
  - **Russian peasant multiplication** is correct *Induction*
  - Largest **n-bit binary** number is $2^n-1$ *Induction, Rosen p. 318*
  - Correctness of **base b conversion** (Algorithm 1 of 4.2), *Strong induction*
  - Size of the **power set** of a finite set with n elements is $2^n$ *Induction, Rosen p. 323*
  - Any int greater than 1 can be written as **product of primes** *Strong induction, Rosen p. 323*
  - There are infinitely many **primes** *Proof by contradiction, Rosen p. 260*
  - **Sum** of geometric progressions $\sum_{j=0}^{n} ar^j = \frac{ar^{n+1} - a}{r - 1}$ when $r \neq 1$, *Induction, Rosen p. 318*
Cautionary tales

• The **basis step** is absolutely necessary … and might need more than one!
• Make sure to stay in the **domain**.
  
  *Recommended practice*
  
  Section 5.1 #49, 50, 51
  
  Section 5.2 #32

• A few **examples** do not guarantee a pattern: cake cutting conundrum. Join all pairs of points among N marked on circumference of cake.
Reminders

• Discussion tomorrow
• HW6 available