Note 3: MATHEMATICAL INDUCTION

- **MATHEMATICAL INDUCTION**
  - used to estabilish that a statement holds for all natural numbers
  - base case, induction hypothesis, inductive step
  - EX: $\sum_{0}^{n} i = \frac{n(n+1)}{2}$
    - PROOF: Prove it by induction
      → base case: $\sum_{0}^{0} i = 0 = \frac{0(0+1)}{2}$
      → inductive step: Suppose the statement holds for some value k, so $\sum_{0}^{k} i = \frac{k(k+1)}{2}$. We wish to prove the statement for $n = k+1$, show $\sum_{0}^{k+1} i = \frac{(k+1)(k+2)}{2}$

$$\sum_{0}^{k+1} i = \sum_{0}^{k} i + (k+1) = \frac{k(k+1)}{2} + k+1 = \frac{(k+1)(k+2)}{2}$$

By principle of mathematical induction, the claim follows

- principle - 2 steps
  1. base case
     → prove that $P(0)$ is true
  2. inductive step
     → show that if $P(k)$ is true, $P(k+1)$ is also true
- EX: Let $x_1, x_2, \dots, x_n$ be real numbers. Then $|x_1 + \dots + x_n| \leq |x_1| + \dots + |x_n|$
  - PROOF: Prove it by induction.
    → base case: $n=1$, then $|x_1| \leq |x_1|$ which is true because they are equal
    → inductive step: Suppose the statement holds for k

$|x_1 + x_2 + \dots + x_{k+1}| \leq |x_1 + \dots + x_k| + |x_{k+1}|$ - triangular inequality

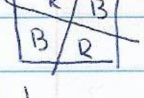
$|x_1 + x_2 + \dots + x_k| \leq |x_1| + \dots + |x_k|$ - hypothesis

Then

$|x_1 + \dots + x_{k+1}| \leq |x_1| + \dots + |x_{k+1}|$

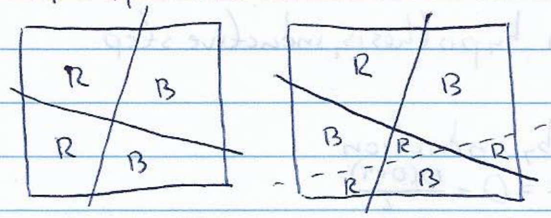
Statement holds for $k+1$. The theorem follows principle of induction.

⇒ **TWO COLOR THEOREM**



- rectangle is divided into regions by drawing lines
- Any map with n lines is two-colorable
  → base case: $n=0$, map with no lines can be colored by one color
  → inductive step: Suppose map with k lines is two-colorable.
  We want to prove that map with $k+1$ lines is two-colorable.
  - note that if we swap red-blue in valid coloring, we will still have a valid coloring

- remove one line from $(k+1)$ map to get map with k lines
  - → by hypothesis it is two-colorable



  - → place the removed line back to get $(k+1)$ line map
    - leave colors on one side unchanged and swap colors on the other side
- why it works
  - consider two regions separated by border, then either
    - → shared border is the line that was removed and replaced (line $k+1$)
      - but by construction, we flipped colors on one side of this line
        - → two regions separated by it have distinct color
    - → shared border is one of the original k lines
      - by hypothesis, the two regions separated by this border have distinct color
  - in both cases, regions separated by border have distinct color

- **STRENGTHENING THE INDUCTION HYPOTHESIS**
  - EX: Sum of the first n odd numbers is a perfect square.
    - ⇒ attempt
      - → base case: first odd number is 1, which is perfect square
      - → inductive step: Suppose the sum of first k numbers is perfect square m^2.
        - the $(k+1)^{st}$ odd number is $2k+1$
        - sum of first $(k+1)$ odd numbers is m^2+2k+1
          - → stuck - induction hypothesis is too weak
  - note that $1: 1 = 1^2$, $2: 1+3 = 4 = 2^2$, $3: 1+3+5 = 9 = 3^2$ } pattern - sum of "n" first odd numbers is n^2

=> prove it by proving a stronger claim

- For all $n \geq 1$, the sum of the first n odd numbers is n^2

- we couldn't prove the original claim, so we hypothesized a stronger one and proved that
  - the original claim was too vague - didn't capture the true structure
    -> the induction hypothesis wasn't enough to prove the desired result
  - new claim is stronger and has structure
- $\boxed{\text{EX}}$: For all natural numbers, $\sum_{i=0}^{n} \frac{1}{2^i} \leq 2$
  - -> prove stronger statement
    - For all natural numbers n, $\sum_{i=0}^{n} \frac{1}{2^i} = 2 - \frac{1}{2^n}$

$$\sum_{i=0}^{k+1} \frac{1}{2^i} = \sum_{i=0}^{k} \frac{1}{2^i} + \frac{1}{2^{k+1}} = \left(2 - \frac{1}{2^k}\right) + \frac{1}{2^{k+1}} = 2 - \left(\frac{1}{2^k} - \frac{1}{2^{k+1}}\right)$$

$$= 2 - \left(\frac{1}{2^{k+1}}\right)$$

- **SIMPLE INDUCTION × STRONG INDUCTION**
  - simple = weak
    - assume statement holds for $P(k)$
  - strong
    - assume statement holds for $P(0), P(1), \ldots, P(k) = \bigwedge_{i=0}^{k} P(i)$ is true
  - equivalent
    - strong induction can't prove statements which weak induction can't
  - $\boxed{\text{EX}}$: The Postage Stamp Problem
    - requires multiple base cases
    - For every natural number $n \geq 12$, it holds that $n = 4x + 5y$ for $x, y \in \mathbb{N}$
    - PROOF
      - -> base case: $n = 12 \to 12 = 4 \cdot 3 + 5 \cdot 0$
        - $n = 13 \to 13 = 4 \cdot 2 + 5 \cdot 1$
        - $n = 14 \to 14 = 4 \cdot 1 + 5 \cdot 2$
        - $n = 15 \to 15 = 4 \cdot 0 + 5 \cdot 3$
      - -> inductive hypothesis: Assume the claim holds for all $12 \leq n \leq k$ for $k \geq 15$
      - -> inductive step: We prove the claim for $n = k+1 \geq 16$
        - note $(k+1) - 4 \geq 12$

- hypothesis implies $(k+1)-4 = 4x' + 5y'$
  - set $x = x'+1$ and $y = y'$
- $\boxed{EX}$: Every natural number has a prime factor.
  - PROOF: Proceed by induction on n.
    - → base case: $P(2)$ holds, 2 is prime and is factor of itself
    - → inductive step: Suppose that for $2 \leq n \leq k$, $P(n)$ holds.
      - show that $P(k+1)$ holds
        - → $k+1$ is composite
          - has factor a such that $1 < a < k+1$
          - $P(a)$ is true so a has some prime factor p
            - → p is then also prime factor of $p+1$
            - → $P(k+1)$ holds
        - → $k+1$ is prime
          - $k+1$ is factor of itself
      - theorem follows by strong induction
- strong and weak induction are equivalent
  - let $Q(n) = P(0) \wedge P(1) \wedge \ldots \wedge P(n)$, then strong induction on P is equivalent to weak induction on Q

• RECURSION AND INDUCTION

⇒ FIBONACCI'S RABBITS

$F(0) = 0$

$F(1) = 1$

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

- we can prove by induction $F(n) \geq 2^{\frac{n-1}{2}}$ for $n \geq 3$
- in recursive function, at least $F(n)$ calls are needed to compute

$F(n)$	$F_2(n)$

function $F(n)$
  if $n=0$:
    return 0
  if $n=0$:
    return 1
  else:
    return $F(n-1) + F(n-2)$
	if $n=0$: return 0
	if $n=1$: return 1
	$a=1, b=0$
	for $k=2$ to n
	  temp$=a$, $a=a+b$, $b=$temp
	return a

- number of iterations of $F_2(n)$ is the same as $F(n)$

⇒ BINARY SEARCH

- input sorted list l and element e
  - → true if element is in list
  - binarySearch(l, e)
    - if len(l) = 1 : return whether $e = l[0]$
    - c = center element of l
    - L = left half of l
    - R = right half of l
    - if $c = e$: return true
    - if $c < e$: return binarySearch(R, e)
    - if $c > e$: return binarySearch(L, e)
- → proof of correctness
  - base case ($n=1$)
    - if l has one element, first statement checks if the element is e and returns correct result
  - inductive step
    - suppose that for all $1 \leq n \leq k$, binarySearch is correct on lists of length n
      - → l doesn't contain e
        - since $e \notin l$, $c \neq e$
        - → two last statements, so we will be recursing on smaller list not containing e
          - since binarySearch is correct, it will return false
      - → l contains e
        - if e is the center element, immidiately return true
        - if $c < e$, e will be right of c
        - if $c > e$, e will be left of c
        - → recurse on appropriate half, that contains c
          - → returns true eventually