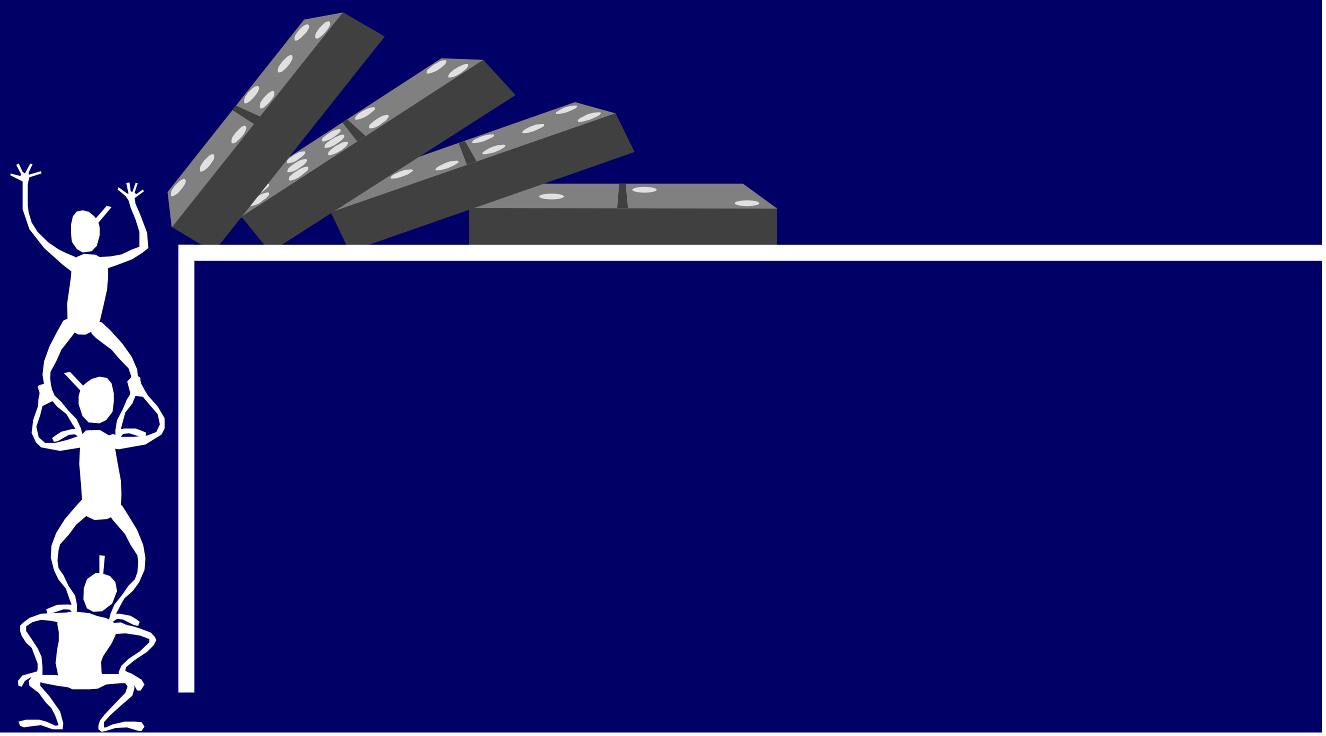
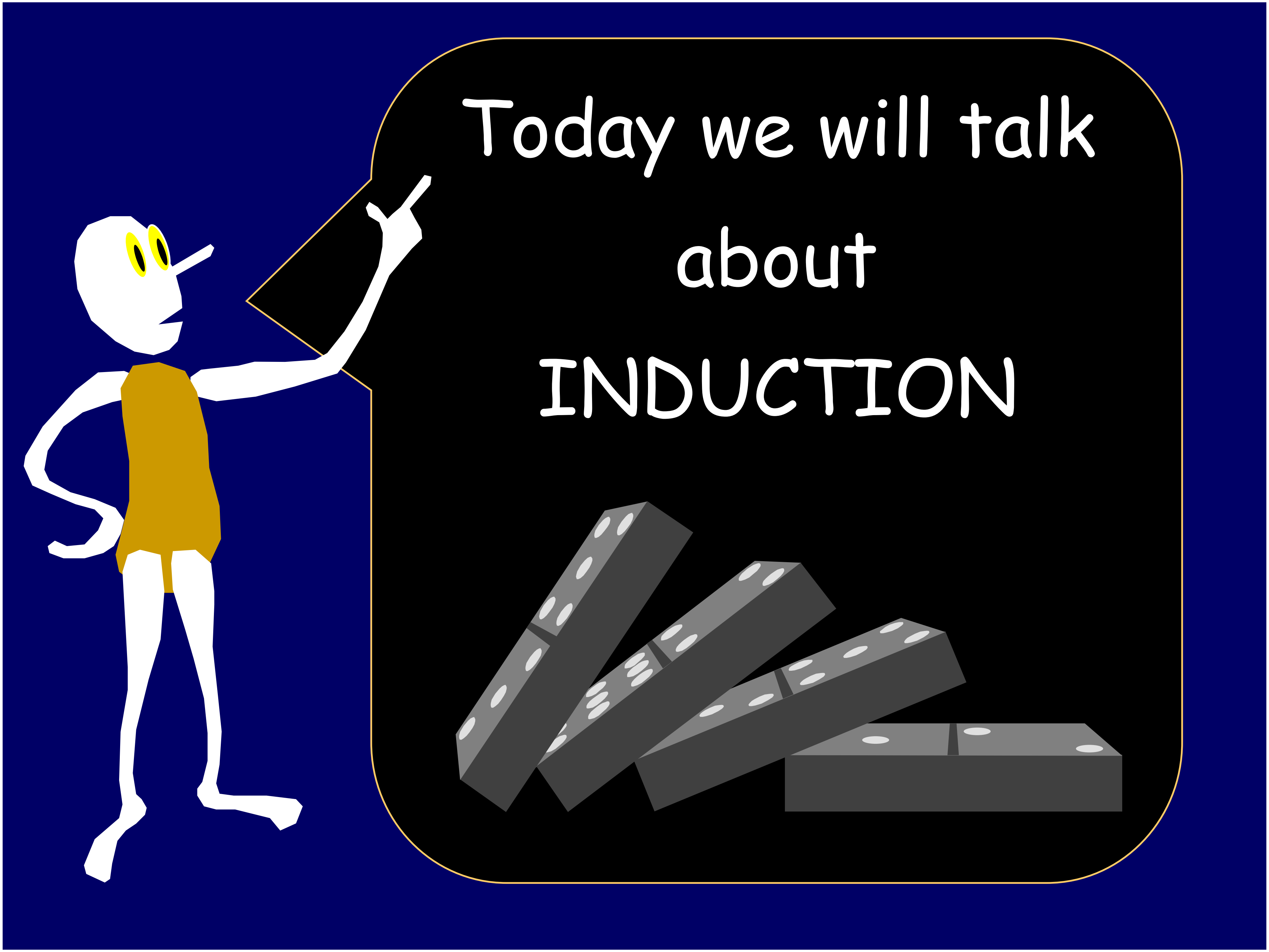
Great Theoretical Ideas In Computer Science

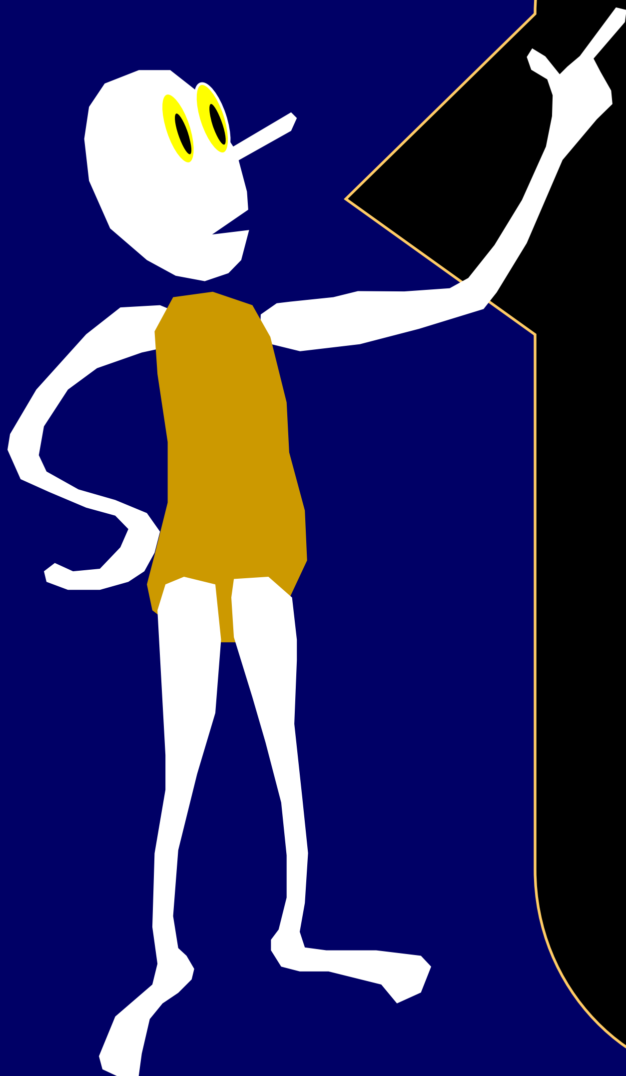
Anupam Gupta CS 15-251 Fall 2006

Lecture 2 Aug 31, 2006 Carnegie Mellon University

Induction: One Step At A Time

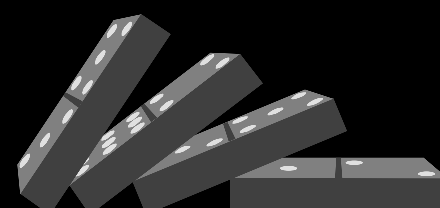



Today we will talk about INDUCTION



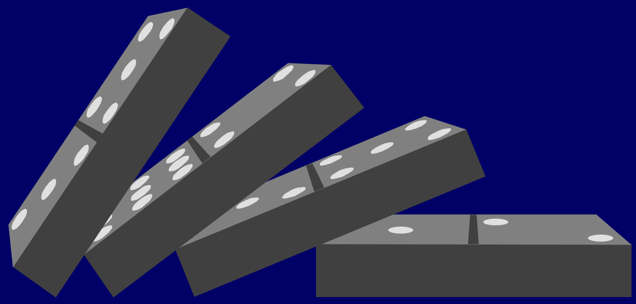
Induction is the primary way we:

1. Prove theorems
2. Construct and define objects



Dominoes

Domino Principle: Line up any number of dominos in a row; knock the first one over and they will all fall



Dominoes Numbered 1 to n

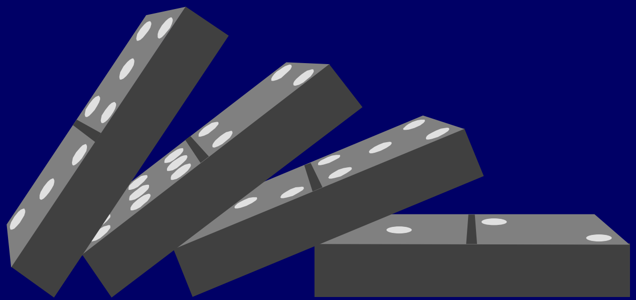
$F_k \equiv$ "The k^{th} domino falls"

If we set them up in a row then each one is set up to knock over the next:

For all $1 \leq k < n$:
$$F_k \Rightarrow F_{k+1}$$

$F_1 \Rightarrow F_2 \Rightarrow F_3 \Rightarrow \ldots$

$F_1 \Rightarrow$ All Dominoes Fall



Standard Notation

"for all" is written "$\forall$"

Example:

For all k>0, P(k) = $\forall k>0, P(k)$

$\exists$ = "there exists"

Dominoes Numbered ~~1~~ 0 to ~~n~~ n-1

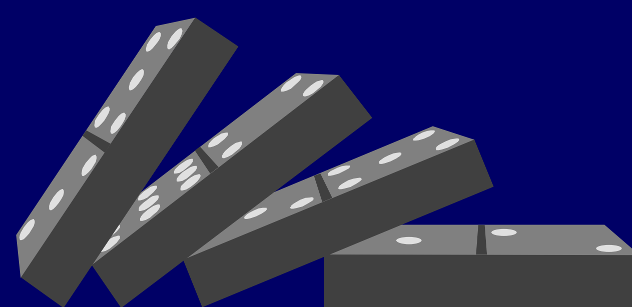
$F_k \equiv$ "The k^{th} domino falls"

$\forall k,\ 0 \leq k < n-1:$

$$F_k \Rightarrow F_{k+1}$$

$F_0 \Rightarrow F_1 \Rightarrow F_2 \Rightarrow \ldots$

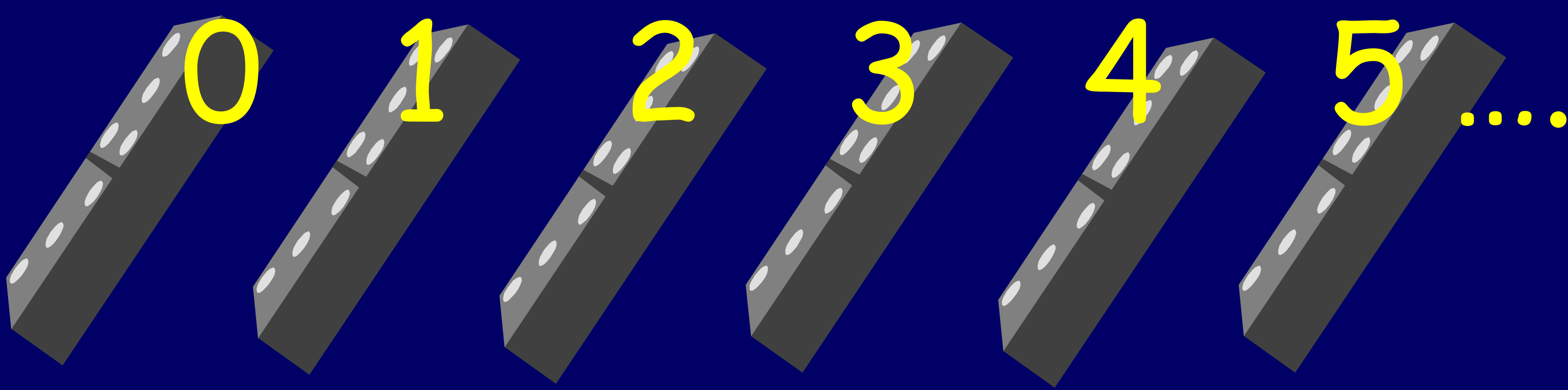
$F_0 \Rightarrow$ All Dominoes Fall

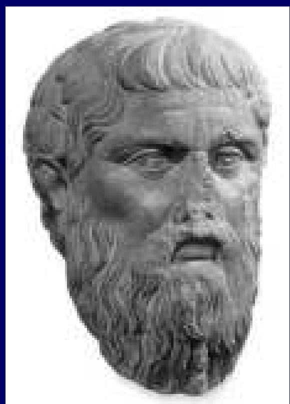


The Natural Numbers

$\mathbb{N}$ = { 0, 1, 2, 3, . . .}

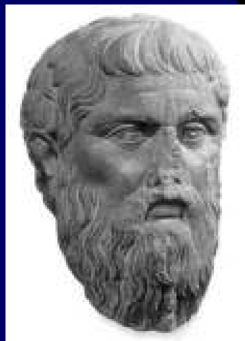
One domino for each natural number:





Plato: The Domino Principle works for an infinite row of dominoes

Aristotle: Never seen an infinite number of anything, much less dominoes.



Plato's Dominoes
One for each natural number

Theorem: An infinite row of dominoes, one domino for each natural number.

Knock over the first domino and they all will fall

Proof:

Suppose they don't all fall. Let $k > 0$ be the lowest numbered domino that remains standing. Domino $k-1 \geq 0$ did fall, but $k-1$ will knock over domino k. Thus, domino k must fall and remain standing. Contradiction.

Mathematical Induction

statements proved instead of dominoes fallen

Infinite sequence of dominoes

$F_k \equiv$ "domino k fell"

Infinite sequence of statements: $S_0, S_1, \ldots$

$F_k \equiv$ "S_k proved"

Establish:
1. F_0
2. For all k, $F_k \Rightarrow F_{k+1}$

Conclude that F_k is true for all k

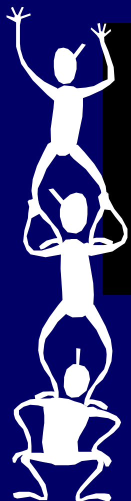
Inductive Proof / Reasoning
To Prove $\forall k \in \mathbb{N}, S_k$

Establish "Base Case": S_0

Establish that $\forall k, S_k \Rightarrow S_{k+1}$

$\forall k, S_k \Rightarrow S_{k+1}$

- Assume hypothetically that S_k for *any* particular k;
- Conclude that S_{k+1}



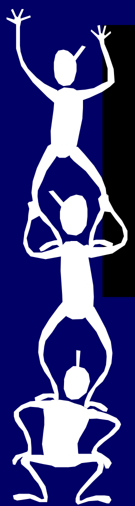
Inductive Proof / Reasoning
To Prove $\forall k \in \mathbb{N}, S_k$

Establish "Base Case": S_0

Establish that $\forall k, S_k \Rightarrow S_{k+1}$

$\forall k, S_k \Rightarrow S_{k+1}$ {

- "Induction Hypothesis" S_k
- "Induction Step"
  Use I.H. to show S_{k+1}



Inductive Proof / Reasoning
To Prove $\forall k \geq b, S_k$

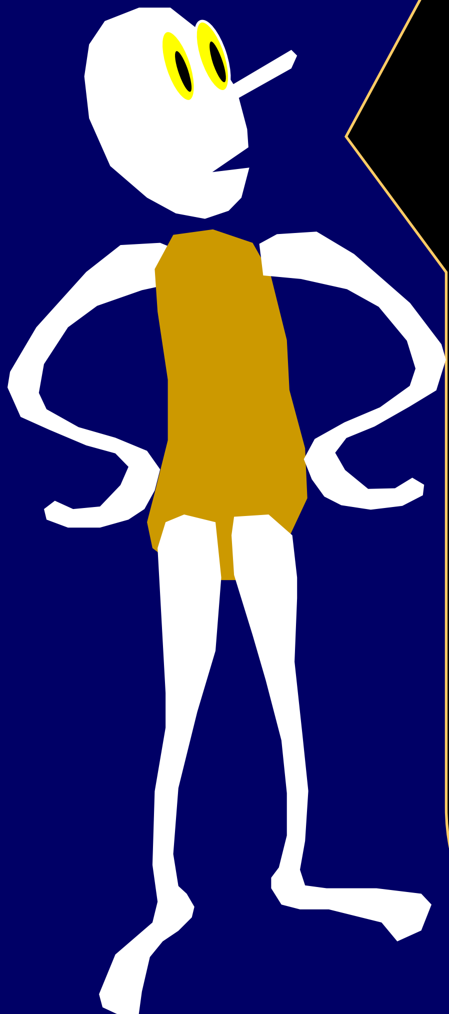
Establish "Base Case": S_b

Establish that $\forall k \geq b, S_k \Rightarrow S_{k+1}$

Assume $k \geq b$

"Inductive Hypothesis": Assume S_k

"Inductive Step:" Prove that S_{k+1} follows

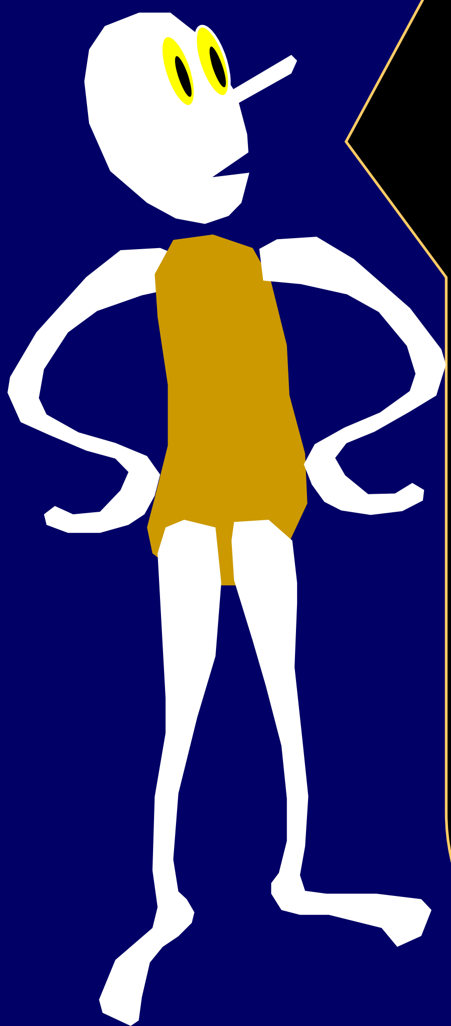


Theorem:?

The sum of the first n odd numbers is n^2.

Check on small values:

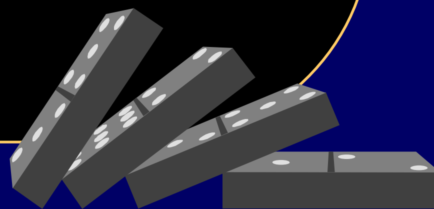
1	= 1
1+3	= 4
1+3+5	= 9
1+3+5+7	= 16

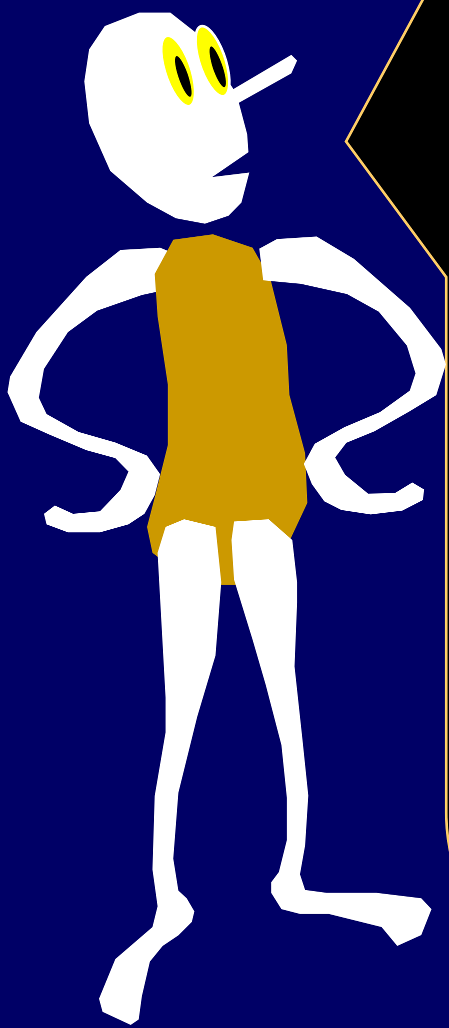


Theorem:?

The sum of the first n odd numbers is n^2.

The k^{th} odd number is expressed by the formula $(2k - 1)$, when $k>0$.



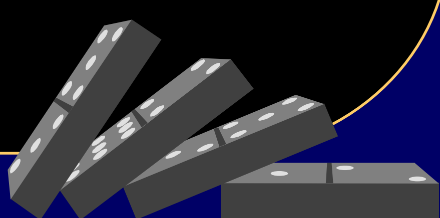


$S_n \equiv$ "The sum of the first n odd numbers is n^2."

Equivalently,

S_n is the statement that:

"$1 + 3 + 5 + (2k-1) + . . +(2n-1) = n^2$"



$S_n \equiv$ "The sum of the first n odd numbers is n^2."

"$1 + 3 + 5 + (2k-1) + . . +(2n-1) = n^2$"

Trying to establish that: $\forall n \geq 1\ S_n$

Base Case: $n=1$ $S_n =$ "$1 = 1^2$" ✓

$\forall k \geq 1 \quad S_k \Rightarrow S_{k+1}$

I.H.: S_k is true i.e. "$1+3+ \cdots +(2k-1) = k^2$"

Inductive Step: $1+3+ \; . \; . \; . + 2k-1 = k^2$

$+(2k+1) \qquad +(2k+1)$

$\Rightarrow 1+3+ \; . \; . \; . +(2k+1) = k^2+2k+1$

$= (k+1)^2$ ☺

$\Rightarrow S_{k+1}$ is true

$S_n \equiv$ "The sum of the first n odd numbers is n^2."

"$1 + 3 + 5 + (2k-1) + . . +(2n-1) = n^2$"

Trying to establish that: $\forall n \geq 1\ S_n$

$S_n \equiv$ "The sum of the first n odd numbers is n^2."

"$1 + 3 + 5 + (2k-1) + . . +(2n-1) = n^2$"

Trying to establish that: $\forall n \geq 1\ S_n$

Assume "Induction Hypothesis": S_k
(for any particular $k \geq 1$)

$1+3+5+\ldots+(2k-1) = k^2$

Induction Step:

Add (2k+1) to both sides.

$1+3+5+\ldots+(2k-1)+(2k+1) = k^2+(2k+1)$

Sum of first k+1 odd numbers $= (k+1)^2$

CONCLUDE: S_{k+1}

$S_n \equiv$ "The sum of the first n odd numbers is n^2."

"$1 + 3 + 5 + (2k-1) + \,.\,.\, +(2n-1) = n^2$"

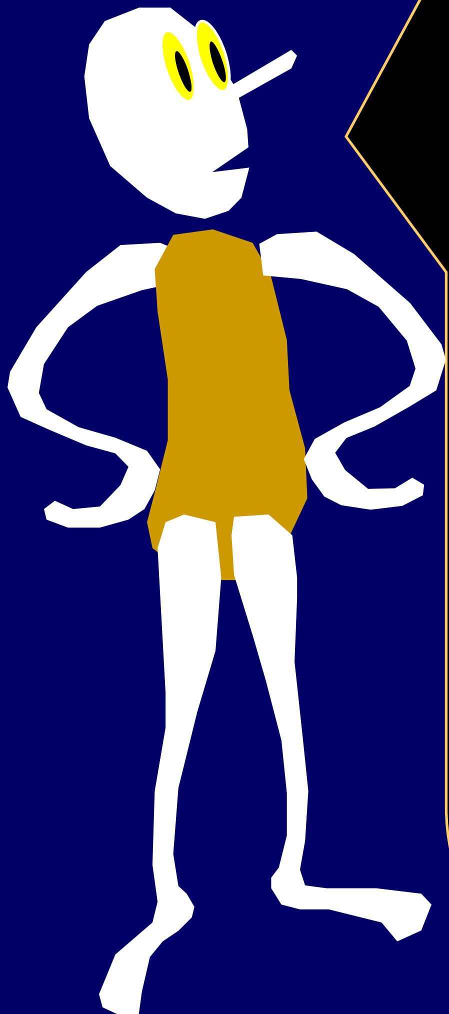
Trying to establish that: $\forall n \geq 1 \; S_n$

In summary:

1) Establish base case: S_1

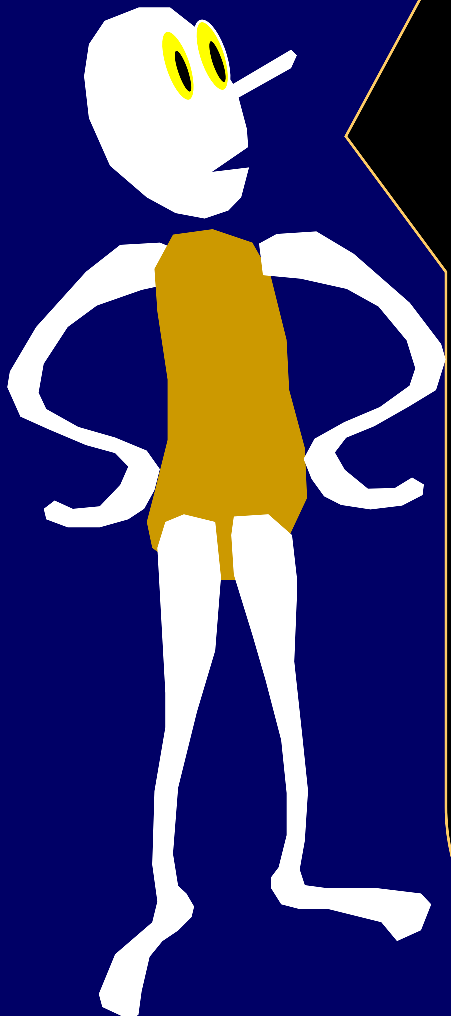
2) Establish domino property: $\forall k \geq 1 \; S_k \Rightarrow S_{k+1}$

By induction on n, we conclude that: $\forall k \geq 1 \; S_k$



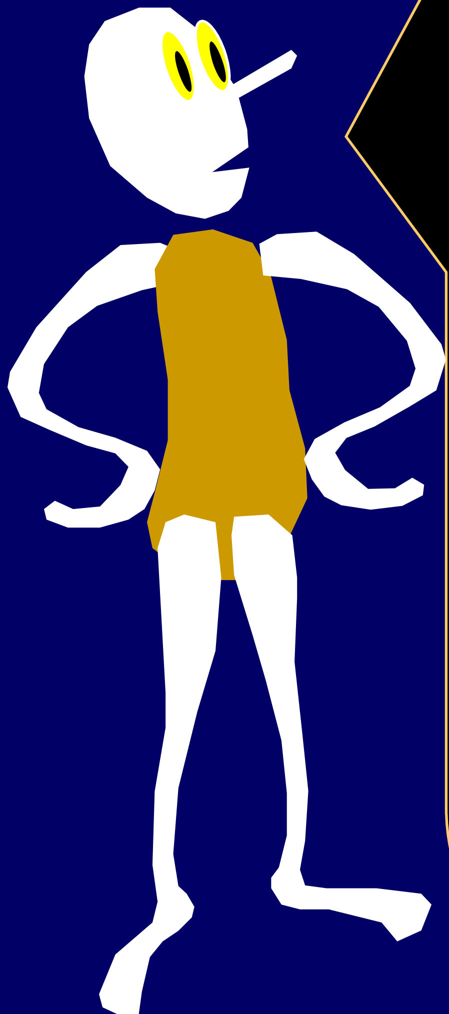
THEOREM:

The sum of the first n odd numbers is n^2.



Theorem?

The sum of the first n numbers is $\frac{1}{2}n(n+1)$.



Theorem? The sum of the first n numbers is $\frac{1}{2}n(n+1)$.

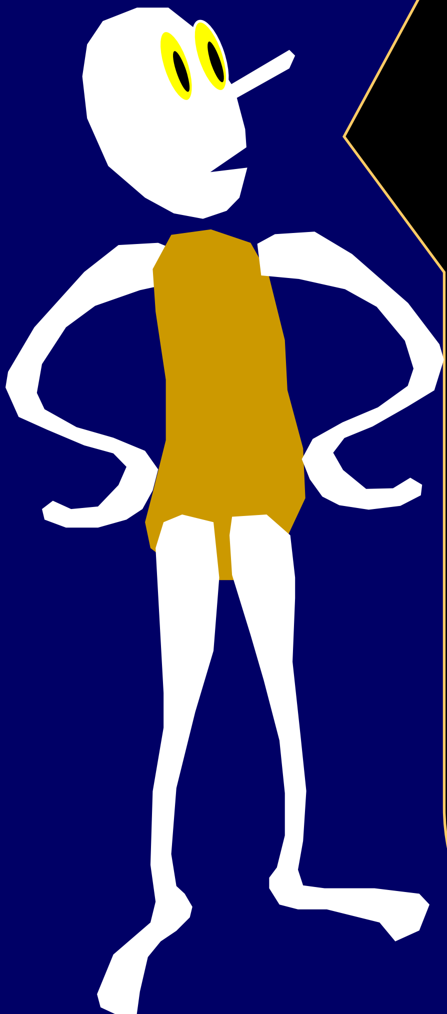
Try it out on small numbers!

$1 = 1 = \frac{1}{2}1(1+1)$.

$1+2 = 3 = \frac{1}{2}2(2+1)$.

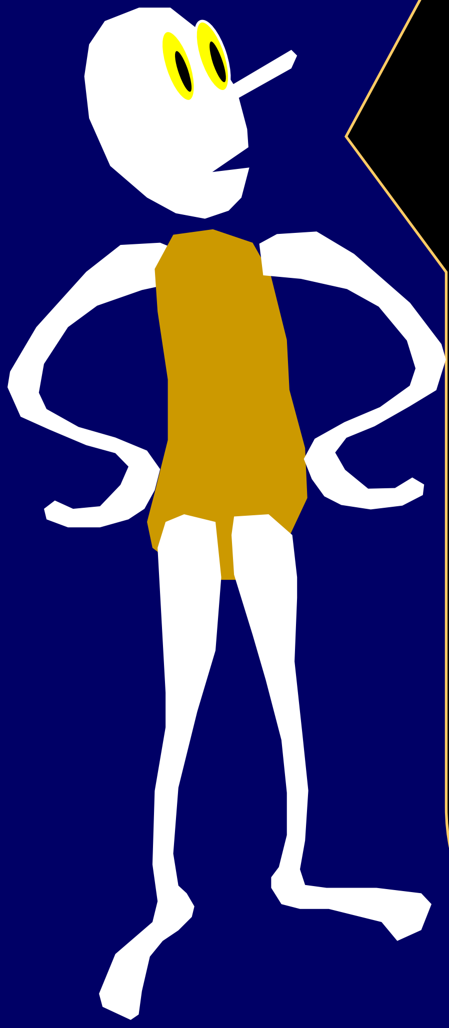
$1+2+3 = 6 = \frac{1}{2}3(3+1)$.

$1+2+3+4 = 10 = \frac{1}{2}4(4+1)$.

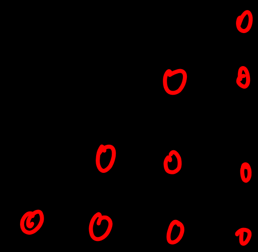


Theorem? The sum of the first n numbers is $\frac{1}{2}n(n+1)$.

	$= 0$	$= \frac{1}{2}0(0+1)$.
1	$= 1$	$= \frac{1}{2}1(1+1)$.
$1+2$	$= 3$	$= \frac{1}{2}2(2+1)$.
$1+2+3$	$= 6$	$= \frac{1}{2}3(3+1)$.
$1+2+3+4$	$= 10$	$= \frac{1}{2}4(4+1)$.



Notation:

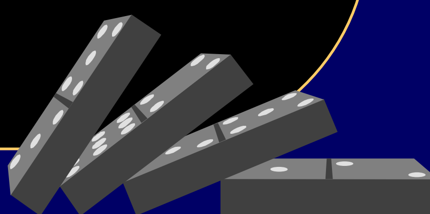


$\Delta_0 = 0$

$\Delta_n = 1 + 2 + 3 + . . . + n-1 + n$

Let S_n be the statement

"$\Delta_n = n(n+1)/2$"



$S_n \equiv$ "$\Delta_n = n(n+1)/2$"

Use induction to prove $\forall k \geq 0, S_k$

Base Case: $\Delta_0 = 0 =$ zero integers

Inductive Hypothesis: $\Delta_k = \frac{k(k+1)}{2}$.

Inductive Step: $\Delta_{k+1} = \frac{(k+1)(k+2)}{2}$ ⇐ assuming S_{k+1}!

DO NOT DO PROOFS THIS WAY!!

$\Leftrightarrow \Delta_k + (k+1) = \frac{k(k+1)}{2} + k+1$

$\Leftarrow \Delta_k = \frac{k(k+1)}{2}$ proving S_k ☹

$S_n \equiv$ "$\Delta_n = n(n+1)/2$"

Use induction to prove $\forall k \geq 0, S_k$

$$\Delta_{k+1} = \Delta_k + (k+1)$$

$$= \frac{k(k+1)}{2} + (k+1) \quad \text{by I.H.}$$

$$= \frac{(k+1)}{2}[k+2] \quad \text{(algebra)}$$

$$= \frac{(k+1)(k+2)}{2}$$

$\Rightarrow S_{k+1}$ is true!

$S_n \equiv$ "$\Delta_n = n(n+1)/2$"

Use induction to prove $\forall k \geq 0, S_k$

Establish "Base Case": S_0.

Δ_0=The sum of the first 0 numbers = 0.

Setting n=0, the formula gives 0(0+1)/2 = 0.

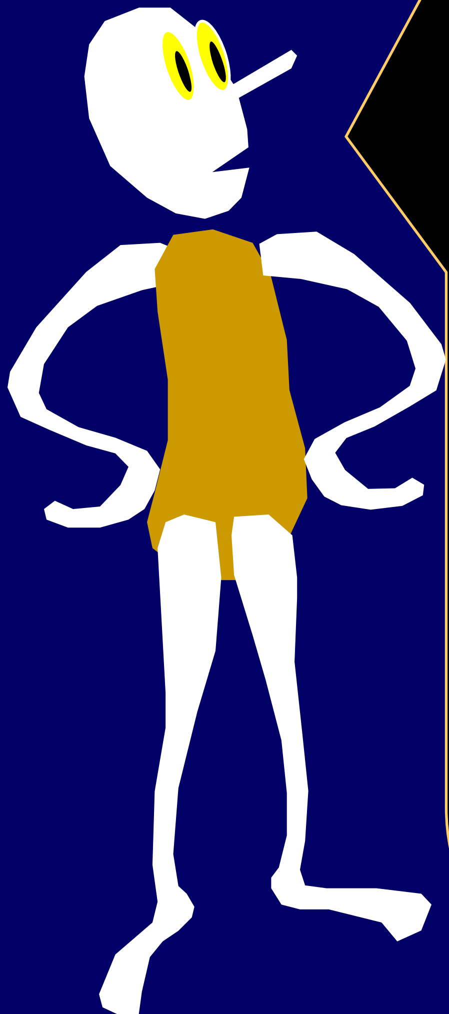
Establish that $\forall k \geq 0, S_k \Rightarrow S_{k+1}$

"Inductive Hypothesis" S_k: $\Delta_k = k(k+1)/2$

$\Delta_{k+1} = \Delta_k + (k+1)$

$= k(k+1)/2 + (k+1)$ [Using I.H.]

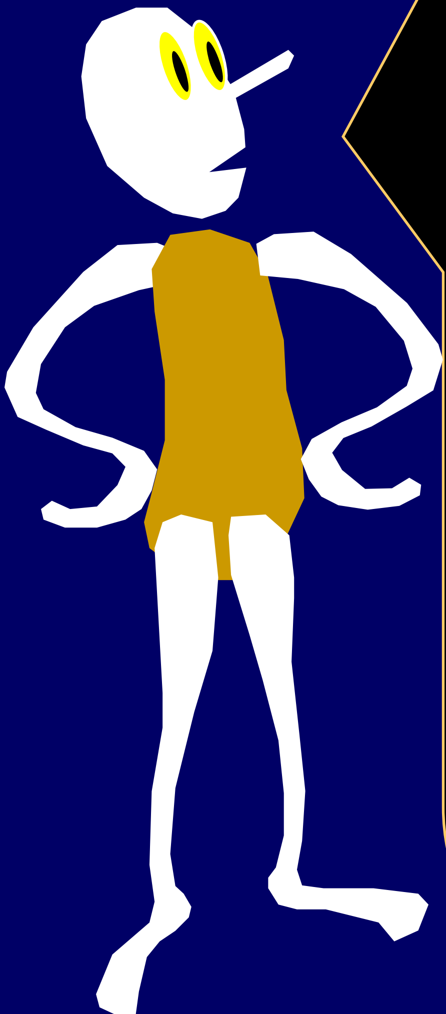
$= (k+1)(k+2)/2$ [which proves S_{k+1}]



Theorem:

The sum of the first n numbers is $\frac{1}{2}n(n+1)$.

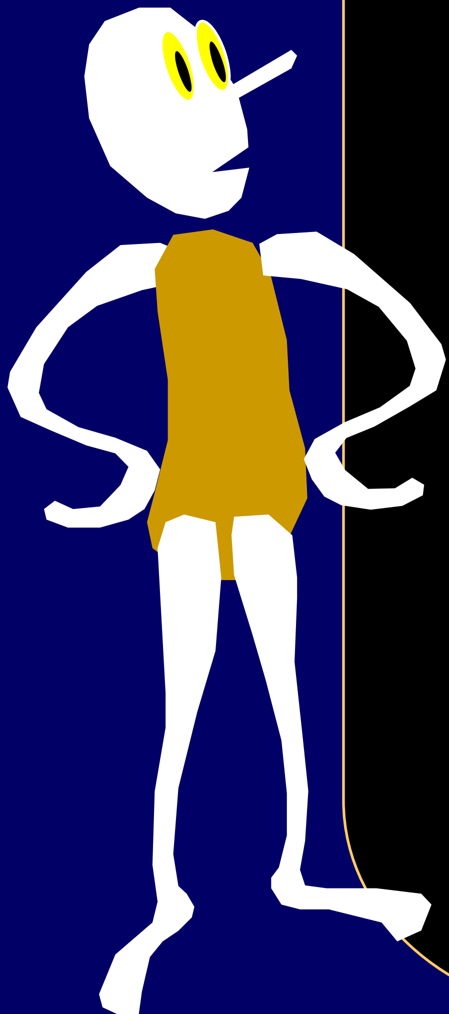
Δ_n



Primes:

A natural number $n>1$ is called **prime** if it has no divisors besides 1 and itself.

n.b. 1 is not considered prime.



Theorem:?

Every natural number > 1 can be factored into primes.

$S_n \equiv$ "n can be factored into primes"

Base case:

2 is prime $\Rightarrow S_2$ is true.

Trying to prove $S_{k-1} \Rightarrow S_k$

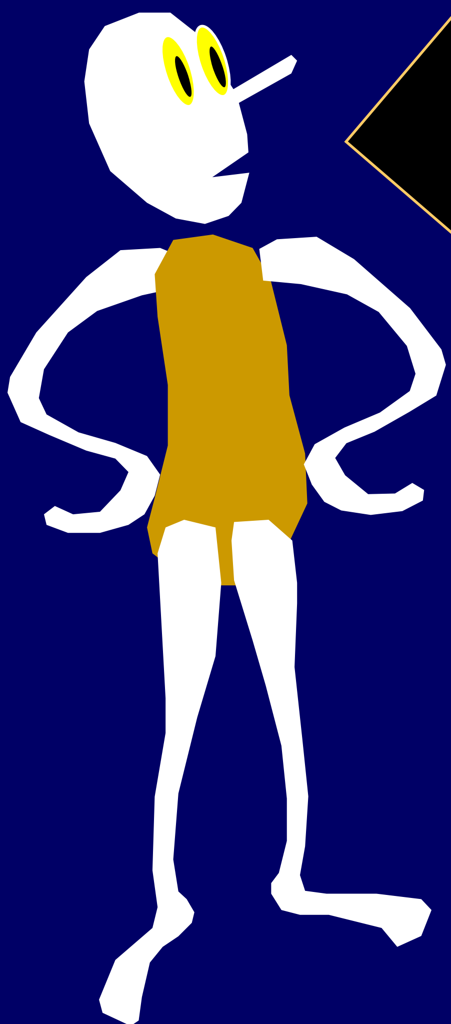
How do we use the fact

$S_{k-1} \equiv$ "k-1 can be factored into primes"

to prove that

$S_k \equiv$ "k can be factored into primes"

Hmm!?



Theorem:?

Every natural number>1 can be factored into primes.

A different approach:

Assume 2,3,…,k-1 all can be factored into primes.

Then show that k can be factored into primes.

$S_n \equiv$ “n can be factored into primes”

Use induction to prove $\forall k > 1$, S_k

$S_n \equiv$ "n can be factored into primes"

Use induction to prove $\forall k > 1, S_k$

Base Case: $2 = 2$.

Inductive hypothesis: $\forall j < k$, S_j is true

(j can be factored into primes)

if k is a prime, ✓

$k = a \cdot b = (p_1 \cdots p_t)(q_1 \cdots q_r)$

All Previous Induction
To Prove $\forall k, S_k$

Also called Strong Induction

Establish Base Case: S_0

Establish that $\forall k, S_k \Rightarrow S_{k+1}$

Let k be any natural number.

Induction Hypothesis:

Assume $\forall j<k, S_j$

Use that to derive S_k

"All Previous" Induction

Repackaged As Standard Induction

Establish Base Case: S_0

Establish Domino Effect:

Let k be any number

Assume $\forall j<k, S_j$

Prove S_k

Define $T_i = \forall j \le i, S_j$

Establish Base Case T_0

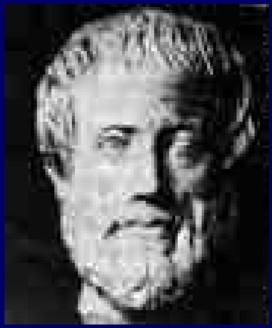
Establish that $\forall k, T_k \Rightarrow T_{k+1}$

Let k be any number

Assume T_{k-1}

Prove T_k


And there are more ways to do inductive proofs



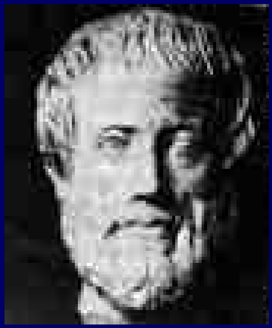
Aristotle's Contrapositive

Let S be a sentence of the form "$A \Rightarrow B$".

The <u>Contrapositive</u> of S is the sentence "$\neg B \Rightarrow \neg A$".

$A \Rightarrow B$: When A is true, B is true.

$\neg B \Rightarrow \neg A$: When B is false, A is false.



Aristotle's Contrapositive

Logically equivalent:

A	B	"$A \Rightarrow B$"	"$\neg B \Rightarrow \neg A$".
False	False	True	True
False	True	True	True
True	False	False	False
True	True	True	True

Contrapositive or Least Counter-Example Induction to Prove $\forall k, S_k$

Establish "Base Case": S_0

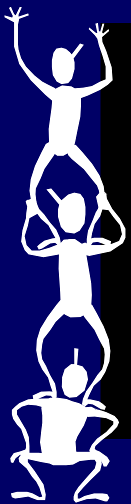
Establish that $\forall k, S_k \Rightarrow S_{k+1}$

$\neg S_{k+1} \Rightarrow \neg S_k$

Let $k>0$ be the least number such that S_k is false.

Prove that $\neg S_k \Rightarrow \neg S_{k-1}$

Contradiction of k being the least counter-example!



Least Counter-Example Induction to Prove $\forall k, S_k$

Establish "Base Case": S_0

Establish that $\forall k, S_k \Rightarrow S_{k+1}$

- Assume that S_k is the least counter-example.
- Derive the existence of a smaller counter-example S_j (for $j < k$)

Rene Descartes [1596-1650]

"Method Of Infinite Decent"

Show that for any counter-example you find a smaller one. Hence, if a counter-example exists there would be an infinite sequence of smaller and smaller counter examples.

Each number > 1 has a prime factorization.

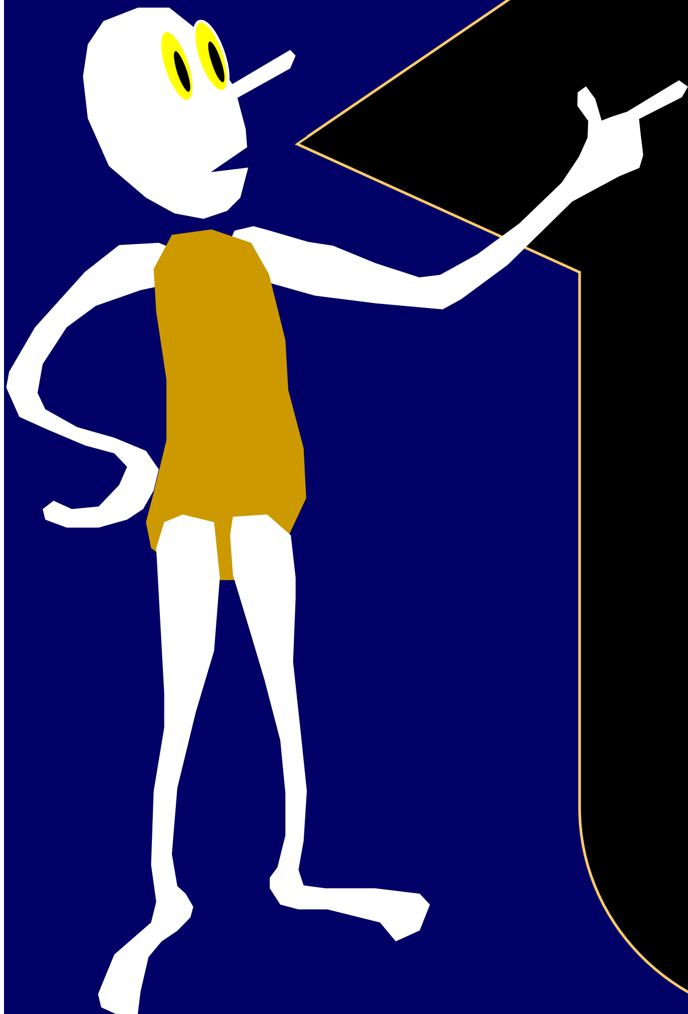
Let n be the least counter-example.

Hence n is not prime

$\Rightarrow$ so $n = ab$.

If both a and b had prime factorizations, then n would too.

Thus a or b is a smaller counter-example.

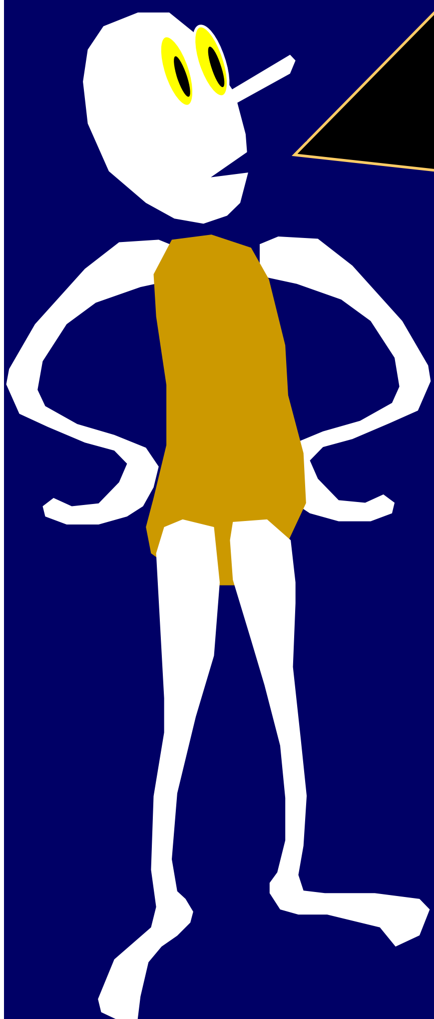


Inductive reasoning is the high level idea:

"Standard" Induction
"All Previous" Induction
"Least Counter-example"
all just
different packaging.

$24 = 2 \times 2 \times 2 \times 3$

$21 = 3 \times 7$



Euclid's theorem on the unique factorization of a number into primes.

Assume there is a least counter-example. Derive a contradiction, or the existence of a smaller counter-example.

Theorem: Each natural has a unique factorization into primes written in non-decreasing order.

$$n = p_1 p_2 \cdots p_r \qquad p_1 \le p_2 \le \cdots \le p_r$$

$$= q_1 q_2 \cdots q_s \qquad q_1 \le q_2 \le \cdots \le q_s$$

Assume: $p_1 > q_1$

$$n \geq p_1 \cdot p_1 \geq ~~p_1 \cdot q_1 + 1~~$$

$$p_1 (q_1 + 1) \geq p_1 \cdot q_1 + 2$$

$$m = n - p_1 q_1 \longleftarrow *$$

$p_1 | m$, $q_1 | m$ $\Rightarrow$ because m has uniq. fact. $m = p_1 \cdot q_1 \cdot z$

Theorem: Each natural has a unique factorization into primes written in non-decreasing order.

$$m = \boxed{p_1 \cdot q_1 \cdot z} = n - p_1 q_1$$

$$= \boxed{p_1 p_2 \cdots p_r - p_1 q_1}$$

dividing by p_1

$$\Rightarrow \quad q_1 \cdot z = (p_2 \cdots p_r - q_1)$$

$$\Rightarrow \quad p_2 \cdots p_r = q_1 \cdot z + q_1 = q_1 (z+1)$$

by uniq fact, q_1 must be one of p_j's.

$\Rightarrow$ contradiction!! ☺

Theorem: Each natural has a unique factorization into primes written in non-decreasing order.

Theorem: Each natural has a unique factorization into primes written in non-decreasing order.

Let n be the least counter-example. n has at least two ways of being written as a product of primes:

$$n = p_1\, p_2 \,..\, p_k = q_1\, q_2 \,...\, q_t$$

The p's must be totally different primes than the q's or else we could divide both sides by one of a common prime and get a smaller counter-example. Without loss of generality, assume $p_1 > q_1$.

Theorem: Each natural has a unique factorization into primes written in non-decreasing order.

Let n be the least counter-example.

$n = p_1 \, p_2 \, .. \, p_k = q_1 \, q_2 \, ... \, q_t$ [$p_1 > q_1$]

$n \geq p_1 p_1 > p_1 \, q_1 + 1$ [Since $p_1 > q_1$]

.

Theorem: Each natural has a unique factorization into primes written in non-decreasing order.

Let n be the least counter-example.

$n = p_1 \, p_2 \, .. \, p_k = q_1 \, q_2 \, ... \, q_t$ [$p_1 > q_1$]

$n \geq p_1 p_1 > p_1 \, q_1 + 1$ [Since $p_1 > q_1$]

$m = n - p_1 q_1$ [Thus $1 < m < n$]

Notice: $m = p_1(p_2 \, .. \, p_k - q_1) = q_1(q_2 \, ... \, q_t - p_1)$

Thus, $p_1|m$ and $q_1|m$

By unique factorization of m, $p_1 q_1|m$, thus $m = p_1 q_1 z$

Theorem: Each natural has a unique factorization into primes written in non-decreasing order.

Let n be the least counter-example.

$n = p_1\, p_2\, ..\, p_k = q_1\, q_2\, ...\, q_t$ [$p_1 > q_1$]

$n \geq p_1 p_1 > p_1\, q_1 + 1$ [Since $p_1 > q_1$]

$m = n - p_1 q_1$ [Thus $1 < m < n$]

Notice: $m = p_1(p_2\, ..\, p_k - q_1) = q_1(q_2\, ...\, q_t - p_1)$

Thus, $p_1|m$ and $q_1|m$

By unique factorization of m, $p_1 q_1|m$, thus $m = p_1 q_1 z$

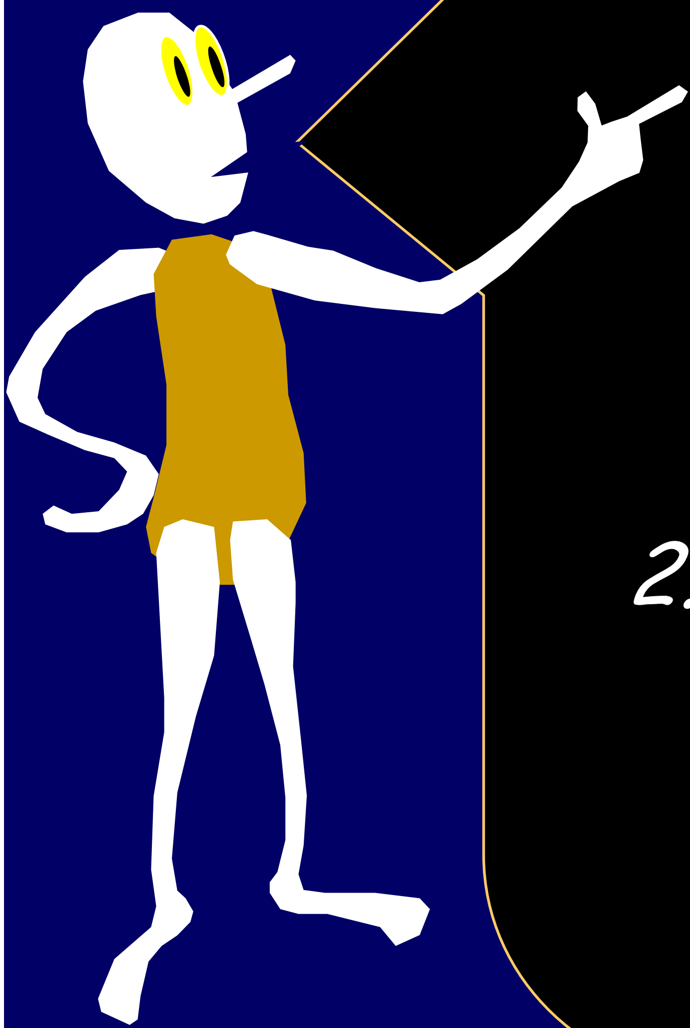
We have: $m = n - p_1 q_1 = p_1(p_2\, ..\, p_k - q_1) = p_1 q_1 z$

Dividing by p_1 we obtain: $(p_2\, ..\, p_k - q_1) = q_1 z$

$p_2\, ..\, p_k = q_1 z + q_1 = q_1(z+1)$ so $q_1|p_2 ... p_k$

And hence, by unique factorization of $p_2 ... p_k$,

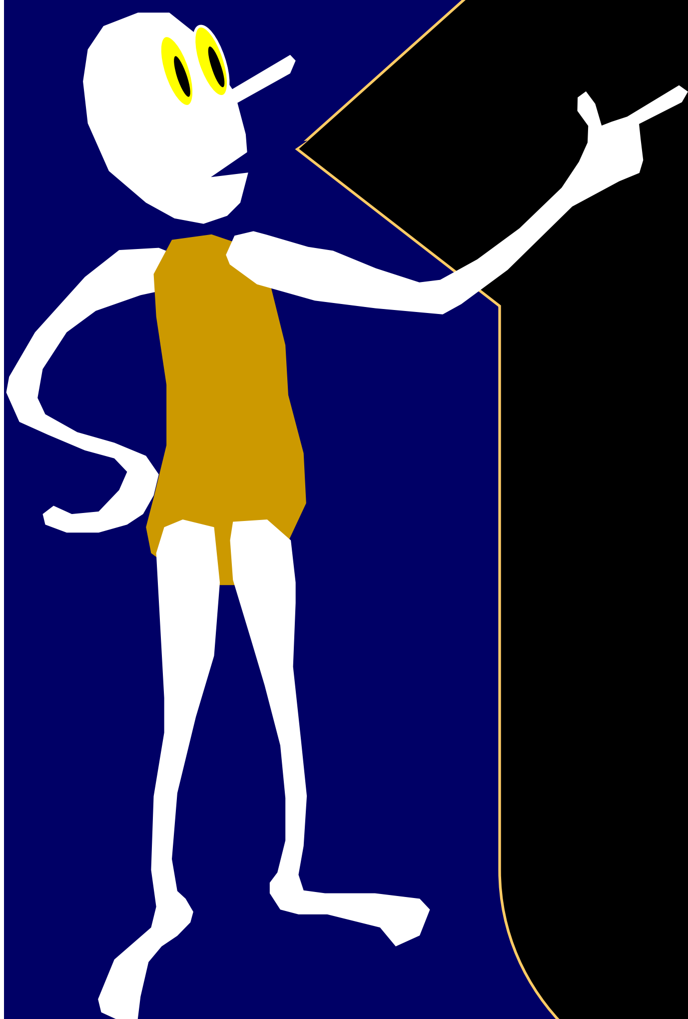
q_1 must be one of the primes $p_2, ..., p_k$. Contradiction of $q_1 < p_1$.



Yet another way of packaging inductive reasoning is to define "invariants".

Invariant:

1. Not varying; constant.
2. <u>*Mathematics.*</u> Unaffected by a designated operation, as a transformation of coordinates.



Yet another way of packaging inductive reasoning is to define "invariants".

Invariant:

3. <u>programming</u> A rule, such as the ordering an ordered list or heap, that applies throughout the life of a data structure or procedure. Each change to the data structure must maintain the correctness of the invariant.

Invariant Induction

Suppose we have a time varying world state: $W_0, W_1, W_2, \ldots$
Each state change is assumed to come from a list of permissible operations. We seek to prove that statement S is true of all future worlds.

Argue that S is true of the initial world.

Show that if S is true of some world – then S remains true after one permissible operation is performed.

Invariant Induction

Suppose we have a time varying world state: $W_0, W_1, W_2, \ldots$

Each state change is assumed to come from a list of permissible operations.

Let S be a statement true of W_0.

Let W be any possible future world state.

Assume S is true of W.

Show that S is true of any world W' obtained by applying a permissible operation to W.

Odd/Even Handshaking Theorem:

At any party at any point in time define a person's parity as ODD/EVEN according to the number of hands they have shaken.

Statement: The number of people of odd parity must be even.

Statement: The number of people of odd parity must be even.

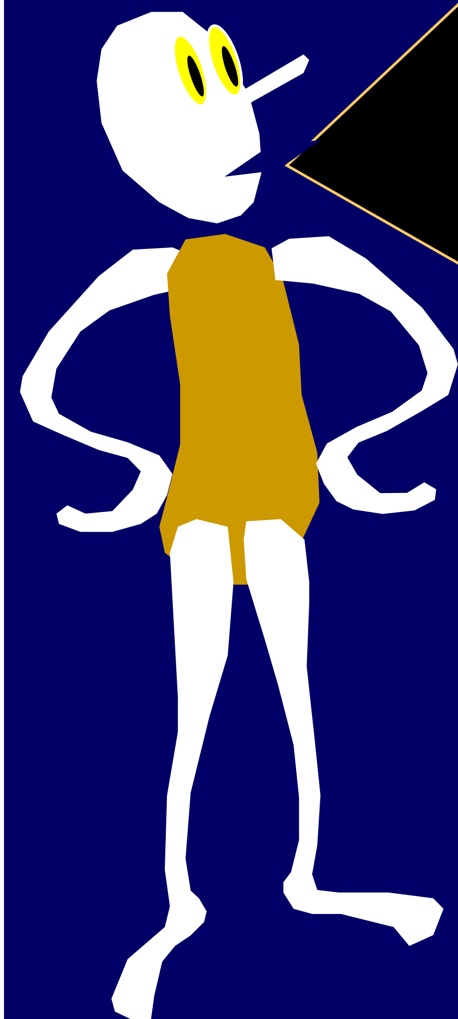
Statement: The number of people of odd parity must be even.

Initial case: Zero hands have been shaken at the start of a party, so zero people have odd parity.

If 2 people of different parities shake, then they both swap parities and the odd parity count is unchanged. If 2 people of the same parity shake, they both change and hence the odd parity count changes by 2 – and remains even.

Inductive reasoning is the high level idea:

Standard Induction
Least Counter-example
All-Previous Induction
Invariants
all just different packaging.



Induction is also how we can define and construct our world.

So many things,
from buildings to computers,
are built up stage by stage,
module by module, each
depending on the previous
stages.

Inductive Definition Of Functions

Stage 0, Initial Condition, or Base Case:
Declare the value of the function on some subset of the domain.

Inductive Rules
Define new values of the function in terms of previously defined values of the function

F(x) is defined if and only if it is implied by finite iteration of the rules.

Inductive Definition

Example

Initial Condition, or Base Case:

F(0) = 1

Inductive definition of the powers of 2!

Inductive Rule:

For n > 0, F(n) = F(n-1) + F(n-1)

n	0	1	2	3	4	5	6	7
F(n)	1	2	4	8	16	32	64	128

Inductive Definition

Example

Initial Condition, or Base Case:

F(1) = 1

F(x) = x for
x being a power of 2!

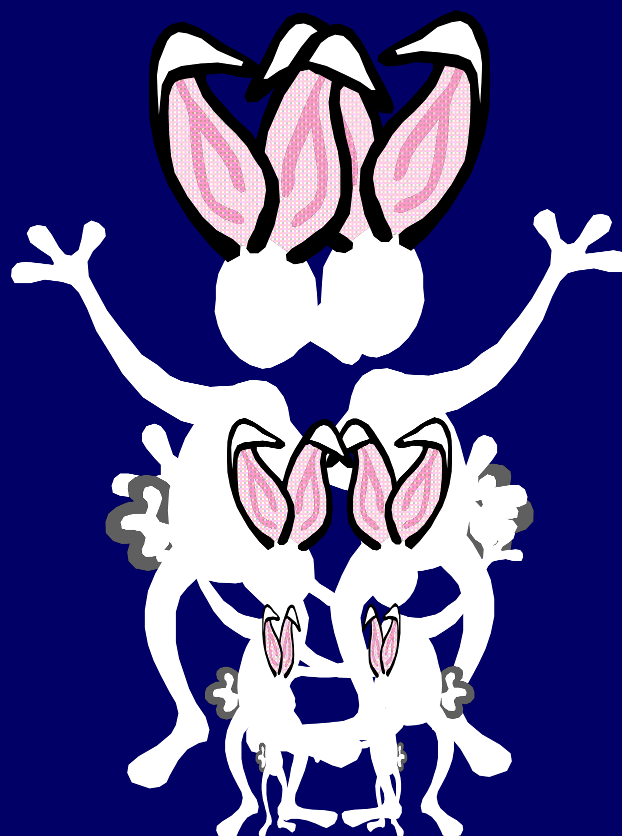
Inductive Rule:

For n > 1, F(n) = F(n/2) + F(n/2)

n	0	1	2	3	4	5	6	7
F(n)	%	1	2	%	4	%	%	%

Leonardo Fibonacci

In 1202, Fibonacci proposed a problem about the growth of rabbit populations



Rabbit Reproduction

A rabbit lives forever

The population starts as single newborn pair

Every month, each productive pair begets a new pair which will become productive after 2 months old

F_n= # of rabbit pairs at the beginning of the n^{th} month

month	1	2	3	4	5	6	7
rabbits	1	1	2	3	5	8	13

Fibonacci Numbers

month	1	2	3	4	5	6	7
rabbits	1	1	2	3	5	8	13

Stage 0, Initial Condition, or Base Case:

Fib(1) = 1; Fib (2) = 1

Inductive Rule:

For $n \geq 3$, Fib(n) = Fib(n-1) + Fib(n-2)

Programs to compute Fib(n)?

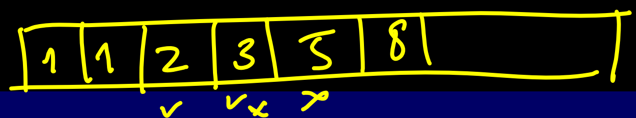
Stage 0, Initial Condition, or Base Case:
Fib(0) = 0; Fib (1) = 1

Inductive Rule
For n>1, Fib(n) = Fib(n-1) + Fib(n-2)

Inductive Definition:
Fib(0)=0, Fib(1)=1, k>1, Fib(k)=Fib(k-1)+Fib(k-2)

Bottom-Up, Iterative Program:

```
Fib(0) = 0; Fib(1) =1;
Input x;
   For k= 2 to x do Fib(k)=Fib(k-1)+Fib(k-2);
Return Fib(x);
```



Top-Down, Recursive Program:

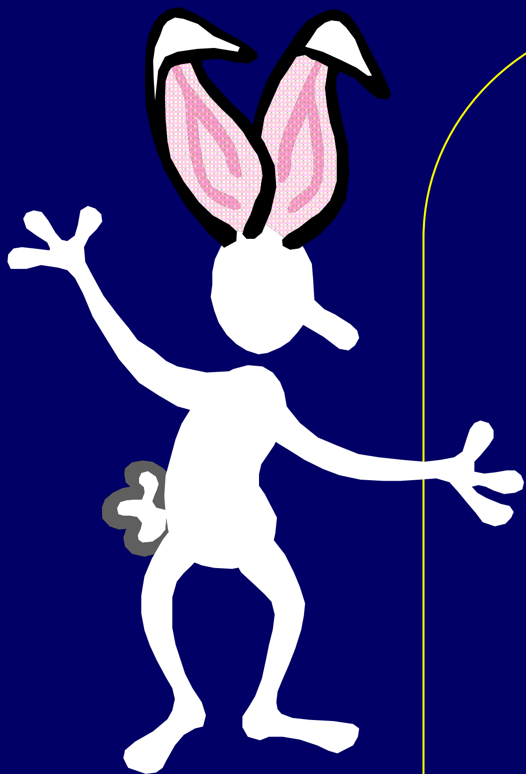
```
Return Fib(x);
Procedure Fib(k)
            If k=0 return 0
                        If k=1 return 1
            Otherwise return Fib(k-1)+Fib(k-2);
```

What is a closed form formula for Fib(n) ????

Stage 0, Initial Condition, or Base Case:
Fib(0) = 0; Fib (1) = 1

Inductive Rule
For n>1, Fib(n) = Fib(n-1) + Fib(n-2)

n	0	1	2	3	4	5	6	7
Fib(n)	0	1	1	2	3	5	8	13



Leonhard Euler (1765)
J. P. M. Binet (1843)
August de Moivre (1730)

$$\mathrm{Fib}(n) = \frac{\left(\frac{\sqrt{5}+1}{2}\right)^{n} - \left(\frac{\sqrt{5}+1}{2}\right)^{-n}}{\sqrt{5}}$$

$$\frac{\phi^{n} - \left(\frac{1}{\phi}\right)^{n}}{\sqrt{5}}$$

Study Bee

Inductive Proof

- Standard Form
- All Previous Form
- Least-Counter Example Form
- Invariant Form

Inductive Definition

- Bottom-Up Programming
- Top-Down Programming
- Recurrence Relations
- Fibonacci Numbers

Logic

- Contrapositive Form of S