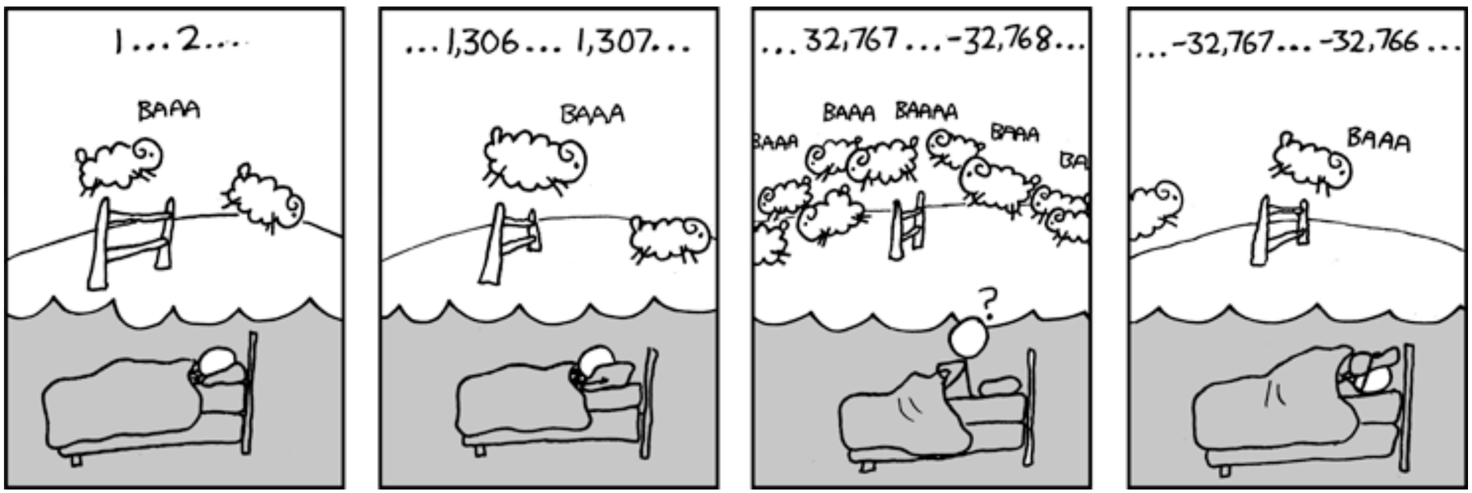
CSE 311: Foundations of Computing

Topic 5: More Number Theory



Administrivia

- HW4 released
  - formal proofs, then translate to English
  - make sure you *understand* the formal proof

    midterm will have formal proofs without Cozy's help

- Warning about CSE cookies...
  - will see Cozy errors if you leave window open for hours

- Added some additional notes on Task 5...
  - "(3x)y" and "3(xy)" are *different* (e.g., produce different code)

    "3xy" means "(3x)y" since left associative

    "xy" is a subexpression of "3(xy)" but not "3xy"

GCD

Recall: Division Theorem

Domain of Discourse
Integers

Division Theorem

For a, b with $b > 0$
there exist *unique* integers q, r with $0 \leq r < b$
such that $a = qb + r$.

To put it another way, if we divide b into a, we get a unique quotient $\boxed{q = a \textbf{ div } b}$
and non-negative remainder $\boxed{r = a \textbf{ mod } b}$

Greatest Common Divisor

Domain of Discourse
Non-negative Integers

GCD Theorem

For a, b with $a > 0$
there exist a *unique* integer n s.t. $n \mid a$ and $n \mid b$
and, for all d, if $d \mid a$ and $d \mid b$, then $d \leq n$

We will denote this unique number as $n = \gcd(a, b)$

Greatest Common Divisor

gcd(a, b):

Largest integer n such that $n \mid a$ and $n \mid b$

- gcd(100, 125) =
- gcd(17, 49) =
- gcd(11, 66) =
- gcd(13, 0) =
- gcd(180, 252) =

Simple GCD fact

Domain of Discourse
Non-negative Integers

Let a be a positive integer.
We have gcd(a, 0) = a.

Simple GCD fact

Domain of Discourse
Non-negative Integers

Let a be a positive integer.
We have gcd(*a*, *0*) = *a*.

Since $0 = 0a$, we can see that $a \mid 0$ by the definition of divides.

Since $a = 1a$, we can see that $a \mid a$ by the definition of divides.

Let d be arbitrary.

Since d was arbitrary, we have shown that a is $\gcd(a, 0)$.

$\gcd(a, 0)$ is the *unique* number n satisfying

$$(n \mid 0) \land (n \mid a) \land \forall d\, (((d \mid 0) \land (d \mid a)) \rightarrow (d \leq n))$$

Simple GCD fact

Domain of Discourse
Non-negative Integers

Let a be a positive integer.
We have gcd(a, 0) = a.

Since $0 = 0a$, we can see that $a \mid 0$ by the definition of divides.

Since $a = 1a$, we can see that $a \mid a$ by the definition of divides.

Let d be arbitrary.

Suppose that $d \mid 0$ and $d \mid a$.

Since d was arbitrary, we have shown that a is $\gcd(a, 0)$.

Simple GCD fact

Domain of Discourse
Non-negative Integers

Let a be a positive integer.
We have gcd(*a*, *0*) = *a*.

Since $0 = 0a$, we can see that $a \mid 0$ by the definition of divides.

Since $a = 1a$, we can see that $a \mid a$ by the definition of divides.

Let d be arbitrary.

Suppose that $d \mid 0$ and $d \mid a$. From the second fact, we get that $a = jd$ for some j by the definition of divides. Then, …

Since d was arbitrary, we have shown that a is $\gcd(a, 0)$.

Simple GCD fact

Domain of Discourse
Non-negative Integers

Let a be a positive integer.
We have gcd(*a*, *0*) = *a*.

Since $0 = 0a$, we can see that $a \mid 0$ by the definition of divides.

Since $a = 1a$, we can see that $a \mid a$ by the definition of divides.

Let d be arbitrary.

Suppose that $d \mid 0$ and $d \mid a$. From the second fact, we get that $a = jd$ for some j by the definition of divides. Since multiplication by non-negative numbers only makes the number bigger, $d \leq a$ holds.

Since d was arbitrary, we have shown that a is $\gcd(a, 0)$.

Oops! This is only true if $j > 0$!

Prop of * $\forall a\, \forall b\, \forall c\, (((a = bc) \land (b > 0)) \rightarrow (c \leq a))$

Simple GCD fact

Domain of Discourse
Non-negative Integers

Let a be a positive integer.
We have gcd(*a, 0*) = *a.*

Since $0 = 0a$, we can see that $a \mid 0$ by the definition of divides.

Since $a = 1a$, we can see that $a \mid a$ by the definition of divides.

Let d be arbitrary. Suppose that $d \mid 0$ and $d \mid a$. From the second fact, we get that $a = jd$ by the definition of divides. We continue by cases…

Suppose that $j > 0$. Then, "Prop of *" tells us that $d \leq a$ holds.

Suppose that $j = 0$.

Since d was arbitrary, we have shown that a is $\gcd(a, 0)$.

Simple GCD fact

Domain of Discourse
Non-negative Integers

Let a be a positive integer.
We have gcd(*a*, *0*) = *a*.

Since $0 = 0a$, we can see that $a \mid 0$ by the definition of divides.

Since $a = 1a$, we can see that $a \mid a$ by the definition of divides.

Let d be arbitrary. Suppose that $d \mid 0$ and $d \mid a$. From the second fact, we get that $a = jd$ by the definition of divides. We continue by cases…

Suppose that $j > 0$. Then, "Prop of *" tells us that $d \leq a$ holds.

Suppose that $j = 0$. That would tell us that $a = 0d = 0$, contradicting the fact that $a > 0$, which was given.

Since d was arbitrary, we have shown that a is $\gcd(a, 0)$.

Simple GCD fact

Domain of Discourse

Non-negative Integers

Let a be a positive integer.
We have gcd(*a, 0*) = *a.*

Since $0 = 0a$, we can see that $a \mid 0$ by the definition of divides.

Since $a = 1a$, we can see that $a \mid a$ by the definition of divides.

Let d be arbitrary. Suppose that $d \mid 0$ and $d \mid a$. From the second fact, we get that $a = jd$ by the definition of divides. We continue by cases…

Suppose that $j > 0$. Then, "Prop of *" tells us that $d \leq a$ holds.

Suppose that $j = 0$. That would tell us that $a = 0d = 0$, contradicting the fact that $a > 0$, which was given. Since false is true, anything is true. In particular, we can say that $d \leq a$ holds.

Since d was arbitrary, we have shown that a is $\gcd(a, 0)$.

Simple GCD fact

Domain of Discourse

Non-negative Integers

Let a be a positive integer.
We have gcd(*a, 0*) = *a.*

Since $0 = 0a$, we can see that $a \mid 0$ by the definition of divides.

Since $a = 1a$, we can see that $a \mid a$ by the definition of divides.

Let d be arbitrary. Suppose that $d \mid 0$ and $d \mid a$. From the second fact, we get that $a = jd$ by the definition of divides. We continue by cases…

Suppose that $j > 0$. Then, "Prop of *" tells us that $d \leq a$ holds.

Suppose that $j = 0$. That would tell us that $a = 0d = 0$, contradicting the fact that $a > 0$, which was given. Since false is true, anything is true. In particular, we can say that $d \leq a$ holds.

Since we have either $j = 0$ or $j > 0$, we see that $d \leq a$ holds in general.

Since d was arbitrary, we have shown that a is $\gcd(a, 0)$.

Useful GCD Fact

Let a and b be positive integers.
We have $\gcd(a, b) = \gcd(b, a \bmod b)$

Proof Idea:
We will show that every number dividing a and b also divides b and $a \bmod b$.
I.e., $d|a$ and $d|b$ iff $d|b$ and $d|(a \bmod b)$.

Hence, their set of common divisors are the same,
which means that their greatest common divisor is the same.

Useful GCD Fact

Let *a* and *b* be positive integers.
We have gcd(*a*, *b*) = gcd(*b*, *a* mod *b*)

Proof (of $d|a$ and $d|b$ iff $d|b$ and $d|(a \bmod b)$)**:**
By the Division Theorem, $a = qb + (a \bmod b)$ for some integer $q = a \operatorname{div} b$.

Suppose $d \mid b$ and $d \mid (a \bmod b)$.
Then $b = md$ and $(a \bmod b) = nd$ for some integers m and n.
Therefore $a = qb + (a \bmod b) = qmd + nd = (qm + n)d$.
So $d \mid a$ by the definition of divides.

Suppose $d \mid a$ and $d \mid b$.
Then $a = kd$ and $b = jd$ for some integers k and j.
Therefore $(a \bmod b) = a - qb = kd - qjd = (k - qj)d$.
So, $d \mid (a \bmod b)$ by the definition of divides.

Since they have the same common divisors, $\gcd(a, b) = \gcd(b, a \bmod b)$. ■

Euclid’s Algorithm

gcd(a, b) = gcd(b, a mod b) gcd(a, 0) = a

```
int gcd(int a, int b) {  /* Assumes: a >= b >= 0 */
    if (b == 0) {
        return a;
    } else {
        return gcd(b, a % b);
    }
}
```

Note: gcd(b, a) = gcd(a, b)

Euclid's Algorithm

Repeatedly use $\gcd(a, b) = \gcd(b, a \bmod b)$ to reduce numbers until you get $\gcd(g, 0) = g$.

gcd(660,126) =

Euclid's Algorithm

Repeatedly use $\gcd(a, b) = \gcd(b, a \bmod b)$ to reduce numbers until you get $\gcd(g, 0) = g$.

$$
\begin{aligned}
\gcd(660,126) &= \gcd(126, 660 \bmod 126) = \gcd(126, 30) \\
&= \gcd(30, 126 \bmod 30) \quad = \gcd(30, 6) \\
&= \gcd(6, 30 \bmod 6) \quad = \gcd(6, 0) \\
&= 6
\end{aligned}
$$

Bézout's theorem

If *a* and *b* are positive integers, then there exist integers ***s*** and ***t*** such that

$$\gcd(a,b) = \boldsymbol{s}a + \boldsymbol{t}b.$$

$$(a > 0 \land b > 0) \rightarrow \exists s\, \exists t\, (\gcd(a,b) = sa + tb)$$

$$\forall a\, \forall b\, ((a > 0 \land b > 0) \rightarrow \exists s\, \exists t\, (\gcd(a,b) = sa + tb))$$

Extended Euclidean algorithm

- Can use Euclid's Algorithm to find s, t such that

$$\gcd(a, b) = sa + tb$$

Extended Euclidean algorithm

- Can use Euclid's Algorithm to find s, t such that

$$\gcd(a, b) = sa + tb$$

Step 1 (Compute GCD & Keep Tableau Information):

$$\gcd(\underset{a}{35}, \underset{b}{27}) = \gcd(\underset{b}{27}, \underset{a \bmod b = r}{35 \bmod 27}) = \gcd(\underset{b}{27}, \underset{r}{8})$$

$$\begin{array}{c} a = q * b + r \\ 35 = 1 * 27 + 8 \end{array}$$

Extended Euclidean algorithm

- Can use Euclid's Algorithm to find s, t such that

$$\gcd(a, b) = sa + tb$$

Step 1 (Compute GCD & Keep Tableau Information):

$$\begin{aligned}
&\overset{a\quad b}{\gcd(35, 27)} = \overset{b\quad a \bmod b = r}{\gcd(27, 35 \bmod 27)} = \overset{b\quad r}{\gcd(27, 8)} \\
&= \gcd(8, 27 \bmod 8) = \gcd(8, 3) \\
&= \gcd(3, 8 \bmod 3) = \gcd(3, 2) \\
&= \gcd(2, 3 \bmod 2) = \gcd(2, 1) \\
&= \gcd(1, 2 \bmod 1) = \gcd(1, 0)
\end{aligned}$$

$$\boxed{\begin{aligned}
a &= q * b + r \\
35 &= 1 * 27 + 8 \\
27 &= 3 * 8 + 3 \\
8 &= 2 * 3 + 2 \\
3 &= 1 * 2 + 1
\end{aligned}}$$

Extended Euclidean algorithm

- Can use Euclid's Algorithm to find s, t such that

$$\gcd(a, b) = sa + tb$$

Step 2 (Solve the equations for r):

$$\boxed{\begin{array}{l} a = q * b + r \\ 35 = 1 * 27 + 8 \\ 27 = 3 * 8 + 3 \\ 8 = 2 * 3 + 2 \\ 3 = 1 * 2 + 1 \end{array}}$$

r = a – q * b

$8 = 35 – 1 * 27$

Extended Euclidean algorithm

- Can use Euclid's Algorithm to find s, t such that

$$\gcd(a, b) = sa + tb$$

Step 2 (Solve the equations for r):

a = q * b + r	r = a – q * b
$35 = 1 * 27 + 8$	$8 = 35 - 1 * 27$
$27 = 3 * 8 + 3$	$3 = 27 - 3 * 8$
$8 = 2 * 3 + 2$	$2 = 8 - 2 * 3$
$3 = 1 * 2 + 1$	$1 = 3 - 1 * 2$

Extended Euclidean algorithm

- **Can use Euclid's Algorithm to find s, t such that**

$$\gcd(a, b) = sa + tb$$

Step 3 (Backward Substitute Equations):

$$8 = 35 - 1 * 27$$

$$3 = 27 - 3 * 8$$

$$2 = 8 - 2 * 3$$

$$\boxed{1 = 3 - 1 * 2}$$

Extended Euclidean algorithm

- Can use Euclid's Algorithm to find s, t such that

$$\gcd(a, b) = sa + tb$$

Step 3 (Backward Substitute Equations):

Plug in the def of 2

$$8 = 35 - 1 * 27$$

$$3 = 27 - 3 * 8$$

$$2 = 8 - 2 * 3$$

$$\boxed{1 = 3 - 1 * 2}$$

$$\begin{aligned} 1 &= 3 - 1 * (8 - 2 * 3) \\ &= 3 - 8 + 2 * 3 \\ &= (-1) * 8 + 3 * 3 \end{aligned}$$

Re-arrange into 3's and 8's

Extended Euclidean algorithm

- Can use Euclid's Algorithm to find s, t such that

$$\gcd(a, b) = sa + tb$$

Step 3 (Backward Substitute Equations):

$$8 = 35 - 1 * 27$$

$$3 = 27 - 3 * 8$$

$$2 = 8 - 2 * 3$$

$$\boxed{1 = 3 - 1 * 2}$$

Plug in the def of 2

$$\begin{aligned} 1 &= 3 - 1 * (8 - 2 * 3) \\ &= 3 - 8 + 2 * 3 \\ &= (-1) * 8 + 3 * 3 \end{aligned}$$

Re-arrange into 3's and 8's

Plug in the def of 3

$$\begin{aligned} &= (-1) * 8 + 3 * (27 - 3 * 8) \\ &= (-1) * 8 + 3 * 27 + (-9) * 8 \\ &= 3 * 27 + (-10) * 8 \end{aligned}$$

Re-arrange into 8's and 27's

Extended Euclidean algorithm

- Can use Euclid's Algorithm to find s, t such that

$$\gcd(a, b) = sa + tb$$

Step 3 (Backward Substitute Equations):

$$8 = 35 - 1 * 27$$

$$3 = 27 - 3 * 8$$

$$2 = 8 - 2 * 3$$

$$\boxed{1 = 3 - 1 * 2}$$

Plug in the def of 2

$$
\begin{aligned}
1 &= 3 - 1 * (8 - 2 * 3) \\
&= 3 - 8 + 2 * 3 \\
&= (-1) * 8 + 3 * 3
\end{aligned}
$$

Re-arrange into 3's and 8's

Plug in the def of 3

$$
\begin{aligned}
&= (-1) * 8 + 3 * (27 - 3 * 8) \\
&= (-1) * 8 + 3 * 27 + (-9) * 8 \\
&= 3 * 27 + (-10) * 8
\end{aligned}
$$

Re-arrange into 8's and 27's

$$
\begin{aligned}
&= 3 * 27 + (-10) * (35 - 1 * 27) \\
&= 3 * 27 + (-10) * 35 + 10 * 27 \\
&= 13 * 27 + (-10) * 35
\end{aligned}
$$

Re-arrange into 27's and 35's

Multiplicative inverse $\mathrm{mod}\ m$

Let $0 \leq a, b < m$. Then, b is the *multiplicative inverse of* a *(modulo* m*)* iff $ab \equiv_m 1$.

x	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6
2	0	2	4	6	1	3	5
3	0	3	6	2	5	1	4
4	0	4	1	5	2	6	3
5	0	5	3	1	6	4	2
6	0	6	5	4	3	2	1

mod 7

x	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9
2	0	2	4	6	8	0	2	4	6	8
3	0	3	6	9	2	5	8	1	4	7
4	0	4	8	2	6	0	4	8	2	6
5	0	5	0	5	0	5	0	5	0	5
6	0	6	2	8	4	0	6	2	8	4
7	0	7	4	1	8	5	2	9	6	3
8	0	8	6	4	2	0	8	6	4	2
9	0	9	8	7	6	5	4	3	2	1

mod 10

Multiplicative inverse $\bmod m$

Suppose $\gcd(a, m) = 1$

By Bézout's Theorem, there exist integers s and t such that $sa + tm = 1$.

s is the multiplicative inverse of a (modulo m):

$1 \equiv_m sa$ since $m \mid 1 - sa$ (since $1 - sa = tm$)

So… we can compute multiplicative inverses with the extended Euclidean algorithm

These inverses let us solve modular equations…

Recall: Properties of Modular Arithmetic

If $\boldsymbol{a} \equiv_{\boldsymbol{m}} \boldsymbol{b}$ and $\boldsymbol{b} \equiv_{\boldsymbol{m}} \boldsymbol{c}$, then $\boldsymbol{a} \equiv_{\boldsymbol{m}} \boldsymbol{c}$.

If $\boldsymbol{a} \equiv_{\boldsymbol{m}} \boldsymbol{b}$, then $\boldsymbol{a} + \boldsymbol{c} \equiv_{\boldsymbol{m}} \boldsymbol{b} + \boldsymbol{c}$.

If $\boldsymbol{a} \equiv_{\boldsymbol{m}} \boldsymbol{b}$, then $\boldsymbol{ac} \equiv_{\boldsymbol{m}} \boldsymbol{bc}$.

These properties are **sufficient** to allow us to do algebra with congruences

In particular, the <u>first two</u> properties let us

- **move** a term from one side to the other
- **simplify** on either side

Recall: Multiplicative inverse $\bmod m$

Suppose $\gcd(a, m) = 1$

By Bézout's Theorem, there exist integers s and t such that $sa + tm = 1$.

s is the multiplicative inverse of a (modulo m):

$1 \equiv_m sa$ since $m \mid 1 - sa$ (since $1 - sa = tm$)

We can compute multiplicative inverses with the Extended Euclidean algorithm

These inverses let us solve modular equations…

Example: Solve a Modular Equation

Solve: $7x \equiv_{26} 3$

Suppose we can show that 15 is the multiplicative inverse of 7 modulo 26, i.e., that $15 \cdot 7 \equiv_{26} 1$

Then, we can multiply on both sides by 15 to see that

$$x \equiv_{26} 1x \equiv_{26} 15 \cdot 7x \equiv_{26} 15 \cdot 3 \equiv_{26} 45 \equiv_{26} 19$$

So, if we are given that $7x \equiv_{26} 3$, then we have shown that $x \equiv_{26} 19$.

Example: Solve a Modular Equation

Solve: $7x \equiv_{26} 3$

Find multiplicative inverse of 7 modulo 26

Example: Solve a Modular Equation

Solve: $7x \equiv_{26} 3$ **Find multiplicative inverse of 7 modulo 26**

$$\gcd(26, 7) = \gcd(7, 5) = \gcd(5, 2) = \gcd(2, 1) = 1$$

$$26 = 3 * 7 + 5$$
$$7 = 1 * 5 + 2$$
$$5 = 2 * 2 + 1$$

Example: Solve a Modular Equation

Solve: $7x \equiv_{26} 3$ **Find multiplicative inverse of 7 modulo 26**

$$\gcd(26,7) = \gcd(7,5) = \gcd(5,2) = \gcd(2,1) = 1$$

$$26 = 3 * 7 + 5 \qquad 5 = 26 - 3 * 7$$

$$7 = 1 * 5 + 2 \qquad 2 = 7 - 1 * 5$$

$$5 = 2 * 2 + 1 \qquad 1 = 5 - 2 * 2$$

Example: Solve a Modular Equation

Solve: $7x \equiv_{26} 3$ **Find multiplicative inverse of 7 modulo 26**

$$\gcd(26,7) = \gcd(7,5) = \gcd(5,2) = \gcd(2,1) = 1$$

$$\begin{aligned} 26 &= 3 * 7 + 5 & \quad 5 &= 26 - 3 * 7 \\ 7 &= 1 * 5 + 2 & \quad 2 &= 7 - 1 * 5 \\ 5 &= 2 * 2 + 1 & \quad 1 &= 5 - 2 * 2 \end{aligned}$$

$$\begin{aligned} 1 &= 5 - 2 * (7 - 1 * 5) \\ &= (-2) * 7 + 3 * 5 \\ &= (-2) * 7 + 3 * (26 - 3 * 7) \\ &= (-11) * 7 + 3 * 26 \end{aligned}$$

Example: Solve a Modular Equation

Solve: $7x \equiv_{26} 3$ **Find multiplicative inverse of 7 modulo 26**

$$\gcd(26,7) = \gcd(7,5) = \gcd(5,2) = \gcd(2,1) = 1$$

$$\begin{array}{ll} 26 = 3 * 7 + 5 & 5 = 26 - 3 * 7 \\ 7 = 1 * 5 + 2 & 2 = 7 - 1 * 5 \\ 5 = 2 * 2 + 1 & 1 = 5 - 2 * 2 \end{array}$$

$$\begin{aligned} 1 &= 5 - 2 * (7 - 1 * 5) \\ &= (-2) * 7 + 3 * 5 \\ &= (-2) * 7 + 3 * (26 - 3 * 7) \\ &= (-11) * 7 + 3 * 26 \end{aligned}$$

Now $(-11) \bmod 26 = 15$. **"the" multiplicative inverse**

(-11 is also "a" multiplicative inverse)

Multiplicative Inverses and Algebra

Adding to both sides easily reversible:

$$\begin{tikzcd} x \equiv_m y \arrow[d, bend left, "+c"] \\ x + c \equiv_m y + c \arrow[u, bend left, "-c"] \end{tikzcd}$$

The same is not true of multiplication…

unless we have a multiplicative inverse $cd \equiv_m 1$

$$\begin{tikzcd} x \equiv_m y \arrow[d, bend left, "\times c"] \\ cx \equiv_m cy \arrow[u, bend left, "\times d"] \end{tikzcd}$$

Example: Solve a Modular Equation

Solve: $7x \equiv_{26} 3$

We saw before that… if we are given that $7x \equiv_{26} 3$, then we have shown that $x \equiv_{26} 19$.

$$7x \equiv_{26} 3 \quad \Rightarrow \quad \mathrm{x} \equiv_{26} 19$$

But these steps are *all* reversible…

Example: Solve a Modular Equation

$$7x \equiv_{26} 3 \quad \Rightarrow \quad 15 \cdot 7\mathrm{x} \equiv_{26} 15 \cdot 3$$

multiply both sides by 15

$$\Rightarrow \quad \mathrm{x} \equiv_{26} 19$$

since $15 \cdot 7 \equiv_{26} 1$ **and** $15 \cdot 3 \equiv_{26} 19$

$$\mathrm{x} \equiv_{26} 19 \quad \Rightarrow \quad 7\mathrm{x} \equiv_{26} 7 \cdot 19$$

multiply both sides by 7

$$\Rightarrow \quad 7\mathrm{x} \equiv_{26} 3$$

since $7 \cdot 19 \equiv_{26} 3$

Example: Solve a Modular Equation

Solve: $7x \equiv_{26} 3$

We saw before that... if we are given that $7x \equiv_{26} 3$, then we have shown that $x \equiv_{26} 19$.

$$7x \equiv_{26} 3 \quad \Rightarrow \quad \mathrm{x} \equiv_{26} 19$$

But all of these steps are reversible...

$$\mathrm{x} \equiv_{26} 19 \quad \Rightarrow \quad 7\mathrm{x} \equiv_{26} 7 \cdot 19$$

So $7x \equiv_{26} 3$ iff $x \equiv_{26} 19$

Hence, the solutions are all numbers of the form $19 + 26k$ for some integer

Solving Modular Equations in "Standard Form"

Solve: $7x \equiv_{26} 3$ (of the form $Ax \equiv_m B$ for some A and B)

Step 1. Find multiplicative inverse of 7 modulo 26

$$1 \;=\; \ldots \;=\; (-11) * 7 \;+\; 3 * 26$$

Since $(-11) \bmod 26 = 15$, the inverse of 7 is 15.

Step 2. Multiply both sides and simplify

Multiplying by 15, we get $x \equiv_{26} 15 \cdot 7x \equiv_{26} 15 \cdot 3 \equiv_{26} 19$.

Step 3. State the full set of solutions

So, the solutions are $19 + 26k$ for any integer k

(must be of the form $a + mk$ with $0 \leq a < m$)

Example Not in "Standard Form"

Solve: $7(x-3) \equiv_{26} 8+2x$

What about equation not in standard form?

Example: Not in "Standard Form"

Solve: $7(x-3) \equiv_{26} 8 + 2x$

Rewrite it in standard form:

$$7x - 21 \equiv_{26} 7(x-3) \equiv_{26} 8 + 2\mathrm{x}$$

move $2\mathrm{x}$ **to the other side**

$$5x - 21 \equiv_{26} 8$$

move -21 **to the other side**

$$5x \equiv_{26} 29 \equiv_{26} 3$$

These steps are all reversible, so the solutions are the same.

Induction

Mathematical Induction

Method for proving claims about non-negative integers

- **A new logical inference rule!**
  - It only applies over the non-negative numbers
  - The idea is to **use** the special structure of these numbers to prove things more easily

Prove $\forall k \; ((a \equiv_m b) \to (a^k \equiv_m b^k))$

Let k be an arbitrary *non-negative* integer.

Suppose that $a \equiv_m b$.

We know $((a \equiv_m b) \wedge (a \equiv_m b)) \to (a^2 \equiv_m b^2)$ **by multiplying congruences. So, applying this repeatedly, we have:**

$$((a \equiv_m b) \wedge (a \equiv_m b)) \to (a^2 \equiv_m b^2)$$
$$\big((a^2 \equiv_m b^2) \wedge (a \equiv_m b)\big) \to (a^3 \equiv_m b^3)$$
$$\dots$$
$$\big((a^{k-1} \equiv_m b^{k-1}) \wedge (a \equiv_m b)\big) \to (a^k \equiv_m b^k)$$

The "…"s is a problem! We don't have a proof rule that allows us to say "do this over and over".

But there is such a rule for non-negative numbers!

Domain: Non-Negative Numbers

Induction
$$\frac{P(0) \qquad \forall k\ (P(k) \longrightarrow P(k+1))}{\therefore \forall n\ \ P(n)}$$

Induction Is A Rule of Inference

Domain: Non-Negative Numbers

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

Induction

How do the givens prove P(3)?

Induction Is A Rule of Inference

Domain: Non-Negative Numbers

Induction

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

How do the givens prove P(5)?

$$P(0) \xrightarrow{P(0)\to P(1)} P(1) \xrightarrow{P(1)\to P(2)} P(2) \xrightarrow{P(2)\to P(3)} P(3) \xrightarrow{P(3)\to P(4\ )} P(4) \xrightarrow{P(4)\to P(5)} P(5)$$

First, we have P(0).
Since P(n) → P(n+1) for all n, we have P(0) → P(1).
  Since P(0) is true and P(0) → P(1), by Modus Ponens, P(1) is true.
Since P(n) → P(n+1) for all n, we have P(1) → P(2).
  Since P(1) is true and P(1) → P(2), by Modus Ponens, P(2) is true.

Using The Induction Rule In A Formal Proof

$$\text{Induction} \quad \frac{P(0) \qquad \forall k\,(P(k) \longrightarrow P(k+1))}{\therefore \forall n\ P(n)}$$

Using The Induction Rule In A Formal Proof

Induction $\dfrac{P(0) \quad \forall k\,(P(k) \longrightarrow P(k+1))}{\therefore \forall n\ P(n)}$

1. P(0)

2. $\forall$k (P(k) $\rightarrow$ P(k+1)) ??
3. $\forall$n P(n) Induction: 1, 2

Using The Induction Rule In A Formal Proof

$$\text{Induction}\quad \frac{P(0) \qquad \forall k\,(P(k) \longrightarrow P(k+1))}{\therefore \forall n\;\; P(n)}$$

1. P(0)

   Let k be an arbitrary integer ≥ 0

   2.1 $P(k) \rightarrow P(k+1)$??

2. $\forall k\,(P(k) \rightarrow P(k+1))$ Intro $\forall$
3. $\forall n\, P(n)$ Induction: 1, 2

Using The Induction Rule In A Formal Proof

$$\text{Induction}\quad \frac{P(0) \qquad \forall k\,(P(k) \longrightarrow P(k+1))}{\therefore \forall n\;\; P(n)}$$

1. P(0)

   Let k be an arbitrary integer ≥ 0

      2.1.1. P(k) — Assumption

      2.1.2. ...

      2.1.3. P(k+1)

   2.1 $P(k) \rightarrow P(k+1)$ — Direct Proof

2. $\forall k\,(P(k) \rightarrow P(k+1))$ — Intro $\forall$
3. $\forall n\, P(n)$ — Induction: 1, 2

Translating to an English Proof

Induction

$$\frac{P(0) \quad \forall k\,(P(k) \longrightarrow P(k+1))}{\therefore \forall n\ P(n)}$$

1. Prove P(0)

Base Case

Let k be an arbitrary integer ≥ 0

2.1.1. Suppose that P(k) is true

Inductive Hypothesis

2.1.2. ...

2.1.3. Prove P(k+1) is true

Inductive Step

2.1 $P(k) \rightarrow P(k+1)$	Direct Proof
2. $\forall k\,(P(k) \rightarrow P(k+1))$	Intro $\forall$
3. $\forall n\ P(n)$	Induction: 1, 2

Conclusion

Translating to an English Proof

1. Prove P(0) **Base Case**

Let k be an arbitrary integer ≥ 0 **Inductive Hypothesis**

2.1.1. Suppose that P(k) is true

2.1.2. ... **Inductive Step**

2.1.3. Prove P(k+1) is true

2.1	$P(k) \rightarrow P(k+1)$	Direct Proof
2.	$\forall k\, (P(k) \rightarrow P(k+1))$	Intro $\forall$
3.	$\forall n\, P(n)$	Induction: 1, 2

Conclusion

Induction English Proof Template

[...Define P(n)...]

We will show that $P(n)$ is true for every $n \geq 0$ by induction.

Base Case: *[...proof of $P(0)$ here...]*

Induction Hypothesis:

Suppose that $P(k)$ is true for an arbitrary $k \geq 0$.

Induction Step:

[...proof of $P(k+1)$ here...]

*The proof of $P(k+1)$ **must** invoke the IH somewhere.*

So, the claim is true by induction.

Inductive Proofs In 5 Easy Steps

Basic induction template

Proof:

1. "Let $P(n)$ be... . We will show that $P(n)$ is true for every $n \geq 0$ by Induction."
2. "Base Case:" Prove $P(0)$
3. "Inductive Hypothesis:

   Suppose $P(k)$ is true for an arbitrary integer $k \geq 0$"
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: Result follows by induction"

What is $1 + 2 + 4 + \ldots + 2^n$?

- $1 \qquad = 1$
- $1 + 2 \qquad = 3$
- $1 + 2 + 4 \qquad = 7$
- $1 + 2 + 4 + 8 \qquad = 15$
- $1 + 2 + 4 + 8 + 16 \qquad = 31$

It sure looks like this sum is $2^{n+1} - 1$

How can we prove it?

> **We could prove it for $n = 1, n = 2, n = 3, \ldots$ but that would literally take forever.**
>
> **Good that we have induction!**

Prove $1 + 2 + 4 + \ldots + 2^n = 2^{n+1} - 1$

Prove $1 + 2 + 4 + \ldots + 2^n = 2^{n+1} - 1$

1. **Let** P(n) **be** “$2^0 + 2^1 + \ldots + 2^n = 2^{n+1} - 1$”. **We will show** P(n) **is true for all non-negative numbers by induction.**

Prove $1 + 2 + 4 + \ldots + 2^n = 2^{n+1} - 1$

1. Let P(n) be "$2^0 + 2^1 + \ldots + 2^n = 2^{n+1} - 1$". **We will show** P(n) **is true for all non-negative numbers by induction.**
2. **Base Case** (n=0): $2^0 = 1 = 2 - 1 = 2^{0+1} - 1$ **so** P(0) **is true.**

Prove $1 + 2 + 4 + \ldots + 2^n = 2^{n+1} - 1$

1. **Let** $P(n)$ **be** "$2^0 + 2^1 + \ldots + 2^n = 2^{n+1} - 1$". **We will show** $P(n)$ **is true for all non-negative numbers by induction.**
2. **Base Case** (n=0): $2^0 = 1 = 2 - 1 = 2^{0+1} - 1$ **so** $P(0)$ **is true.**
3. **Induction Hypothesis: Suppose that** $P(k)$ **is true for some arbitrary integer** $k \geq 0$**, i.e., that** $\boxed{2^0 + 2^1 + \ldots + 2^k = 2^{k+1} - 1}$**.**

Prove $1 + 2 + 4 + \ldots + 2^n = 2^{n+1} - 1$

1. **Let** $P(n)$ **be** "$2^0 + 2^1 + \ldots + 2^n = 2^{n+1} - 1$". **We will show** $P(n)$ **is true for all non-negative numbers by induction.**
2. **Base Case** (n=0): $2^0 = 1 = 2 - 1 = 2^{0+1} - 1$ **so** $P(0)$ **is true.**
3. **Induction Hypothesis: Suppose that** $P(k)$ **is true for some arbitrary integer** $k \geq 0$**, i.e., that** $2^0 + 2^1 + \ldots + 2^k = 2^{k+1} - 1$**.**
4. **Induction Step:**

   Goal: Show $P(k+1)$**, i.e. show** $2^0 + 2^1 + \ldots + 2^k + 2^{k+1} = 2^{k+2} - 1$

Prove $1 + 2 + 4 + \ldots + 2^n = 2^{n+1} - 1$

1. **Let** P(n) **be** "$2^0 + 2^1 + \ldots + 2^n = 2^{n+1} - 1$". **We will show** P(n) **is true for all non-negative numbers by induction.**
2. **Base Case** (n=0): $2^0 = 1 = 2 - 1 = 2^{0+1} - 1$ **so** P(0) **is true.**
3. **Induction Hypothesis: Suppose that** P(k) **is true for some arbitrary integer** $k \geq 0$, **i.e., that** $2^0 + 2^1 + \ldots + 2^k = 2^{k+1} - 1$.
4. **Induction Step:**

$$2^0 + 2^1 + \ldots + 2^k = 2^{k+1} - 1 \quad \textbf{by IH}$$

Adding 2^{k+1} **to both sides, we get:**

$$2^0 + 2^1 + \ldots + 2^k + 2^{k+1} = 2^{k+1} + 2^{k+1} - 1$$

Note that $2^{k+1} + 2^{k+1} = 2(2^{k+1}) = 2^{k+2}$.

So, we have $2^0 + 2^1 + \ldots + 2^k + 2^{k+1} = 2^{k+2} - 1$, **which is exactly** P(k+1).

Recall: Substitution vs Adding Equations

If $\boldsymbol{a} = \boldsymbol{b}$ and $\boldsymbol{b} = \boldsymbol{c}$, then $\boldsymbol{a} = \boldsymbol{c}$	"Transitivity"
If $\boldsymbol{a} = \boldsymbol{b}$ and $\boldsymbol{c} = \boldsymbol{d}$, then $\boldsymbol{a} + \boldsymbol{c} = \boldsymbol{b} + \boldsymbol{d}$	"Add Equations"
If $\boldsymbol{a} = \boldsymbol{b}$ and $\boldsymbol{c} = \boldsymbol{d}$, then $\boldsymbol{ac} = \boldsymbol{bd}$	"Multiply Equations"

- **Substitution is an alternative for solving problems**
  - **we will try this out on HW4**
  - **will be heavily used in *future* homework**

Recall: Equivalence Chains

$p \land (p \to r) \equiv p \land (\neg p \lor r)$	Law of Implication
$\equiv (p \land \neg p) \lor (p \land r)$	Distributive
$\equiv \mathbf{F} \lor (p \land r)$	Negation
$\equiv (p \land r) \lor \mathbf{F}$	Commutative
$\equiv p \land r$	Identity

- Each line explains equivalence with *previous line*
  - e.g., $(p \land r) \lor F \equiv p \land r$ by Identity

- Entire chain proves $p \land (p \to r) \equiv p \land r$
  - follows by transitivity of "$\equiv$"

Calculation Block

We can do the same with equality:

$$\begin{aligned} & 2^0 + 2^1 + \ldots + 2^k + 2^{k+1} \\ &\quad = (2^0+2^1+ \ldots + 2^k) + 2^{k+1} \\ &\quad = (2^{k+1} - 1) + 2^{k+1} && \text{since } 2^0+2^1+ \ldots + 2^k = 2^{k+1} - 1 \\ &\quad = 2(2^{k+1}) - 1 \\ &\quad = 2^{k+2} - 1 \end{aligned}$$

Explanations appear on in right column

- "since" means we substituted LHS for RHS
- ordinary algebra (on integers) does not need explanation
- "def of" will be used to apply the definition of a function

  e.g., replacing f(x) by y when we have f defined as f(x) := y

Calculation Block

We can do the same with equality:

$$2^0 + 2^1 + \ldots + 2^k + 2^{k+1}$$
$$= (2^0+2^1+ \ldots + 2^k) + 2^{k+1}$$
$$= (2^{k+1} – 1) + 2^{k+1} \qquad \text{since } 2^0+2^1+ \ldots + 2^k = 2^{k+1} – 1$$
$$= 2(2^{k+1}) – 1$$
$$= 2^{k+2} – 1$$

Entire block shows $2^0 + 2^1 + \ldots + 2^k + 2^{k+1} = 2^{k+2} – 1$

– **this is the transitivity property of "="**

Can also do calculation with "<" and "≤"

– **don't mix directions: ">" and "<" in one block is ><**

Prove $1 + 2 + 4 + \ldots + 2^n = 2^{n+1} - 1$

1. **Let** P(n) **be** "$2^0 + 2^1 + \ldots + 2^n = 2^{n+1} - 1$". **We will show** P(n) **is true for all non-negative numbers by induction.**
2. **Base Case** (n=0): $2^0 = 1 = 2 - 1 = 2^{0+1} - 1$ **so** P(0) **is true.**
3. **Induction Hypothesis: Suppose that** P(k) **is true for some arbitrary integer** $k \geq 0$**, i.e., that** $2^0 + 2^1 + \ldots + 2^k = 2^{k+1} - 1$**.**
4. **Induction Step:**

   We can calculate

$$\begin{aligned} 2^0 + 2^1 + \ldots + 2^k + 2^{k+1} &= (2^0 + 2^1 + \ldots + 2^k) + 2^{k+1} \\ &= (2^{k+1} - 1) + 2^{k+1} \qquad \textbf{by the IH} \\ &= 2(2^{k+1}) - 1 \\ &= 2^{k+2} - 1, \end{aligned}$$

   which is exactly P(k+1)**.**

The entire inductive step is one calculation!

We will rely heavily on calculation going forward...

Prove $1 + 2 + 4 + \ldots + 2^n = 2^{n+1} - 1$

1. **Let** P(n) **be** "$2^0 + 2^1 + \ldots + 2^n = 2^{n+1} - 1$". **We will show** P(n) **is true for all non-negative numbers by induction.**
2. **Base Case** (n=0): $2^0 = 1 = 2 - 1 = 2^{0+1} - 1$ **so** P(0) **is true.**
3. **Induction Hypothesis: Suppose that** P(k) **is true for some arbitrary integer** $k \geq 0$**, i.e., that** $2^0 + 2^1 + \ldots + 2^k = 2^{k+1} - 1$.
4. **Induction Step:**

   We can calculate

$$
\begin{aligned}
2^0 + 2^1 + \ldots + 2^k + 2^{k+1} &= (2^0+2^1+ \ldots + 2^k) + 2^{k+1} \\
&= (2^{k+1} - 1) + 2^{k+1} \qquad \textbf{by the IH} \\
&= 2(2^{k+1}) - 1 \\
&= 2^{k+2} - 1,
\end{aligned}
$$

   which is exactly P(k+1).

5. **Thus** P(n) **is true for all** n **≥ 0, by induction.**

Prove $1 + 2 + 3 + \ldots + n = n(n+1)/2$

Prove that $\sum_{i=0}^{n} i = n(n+1)/2$

Summation Notation

$$\sum_{i=0}^{n} i = 0 + 1 + 2 + 3 + \ldots + n$$

Prove $\sum_{i=0}^{n} i = n(n+1)/2$

1. Let P(n) be "$\sum_{i=0}^{n} i = n(n+1)/2$". We will show P(n) is true for all non-negative numbers by induction.

Prove $\sum_{i=0}^{n} i = n(n+1)/2$

1. **Let** P(n) **be** "$\sum_{i=0}^{n} i = n(n+1)/2$". **We will show** P(n) **is true for all non-negative numbers by induction.**
2. **Base Case** (n=0): $\sum_{i=0}^{0} i = 0 = 0(0+1)/2$, so P(0) **is true.**

Prove $\sum_{i=0}^{n} i = n(n+1)/2$

1. **Let** P(n) **be** "$\sum_{i=0}^{n} i = n(n+1)/2$". **We will show** P(n) **is true for all non-negative numbers by induction.**
2. **Base Case** (n=0): $\sum_{i=0}^{0} i = 0 = 0(0+1)/2$, so P(0) **is true.**
3. **Induction Hypothesis: Suppose that** P(k) **is true for some arbitrary integer** k ≥ 0**. I.e., suppose** $\sum_{i=0}^{k} i = k(k+1)/2$

"some" or "an"
not <u>any</u>!

Prove $\sum_{i=0}^{n} i = n(n+1)/2$

1. **Let** P(n) **be** "$\sum_{i=0}^{n} i = n(n+1)/2$". **We will show** P(n) **is true for all non-negative numbers by induction.**
2. **Base Case** (n=0): $\sum_{i=0}^{0} i = 0 = 0(0+1)/2$, so P(0) **is true.**
3. **Induction Hypothesis: Suppose that** P(k) **is true for some arbitrary integer** k ≥ 0. **I.e., suppose** $\sum_{i=0}^{k} i = k(k+1)/2$
4. **Induction Step:**

Goal: Show $\mathrm{P}(\mathrm{k}+1)$, **i.e.,** $\sum_{i=0}^{k+1} i = (k+1)(k+2)/2$

Prove $\sum_{i=0}^{n} i = n(n+1)/2$

1. **Let** P(n) **be** "$\sum_{i=0}^{n} i = n(n+1)/2$". **We will show** P(n) **is true for all non-negative numbers by induction.**
2. **Base Case** (n=0): $\sum_{i=0}^{0} i = 0 = 0(0+1)/2,$ so P(0) **is true.**
3. **Induction Hypothesis: Suppose that** P(k) **is true for some arbitrary integer** k ≥ 0**. I.e., suppose** $\sum_{i=0}^{k} i = k(k+1)/2$
4. **Induction Step: We can see that**

$$\begin{aligned}
\sum_{i=0}^{k+1} i &= \left(\sum_{i=0}^{k} i\right) + (k+1) \\
&= \mathrm{k}(\mathrm{k}+1)/2 + (k+1) && \textbf{by IH} \\
&= (\mathrm{k}+1)(\mathrm{k}/2+1) \\
&= (\mathrm{k}+1)(\mathrm{k}+2)/2
\end{aligned}$$

which is exactly P(k+1)**.**

Prove $\sum_{i=0}^{n} i = n(n+1)/2$

1. **Let** P(n) **be** "$\sum_{i=0}^{n} i = n(n+1)/2$". **We will show** P(n) **is true for all non-negative numbers by induction.**
2. **Base Case** (n=0): $\sum_{i=0}^{0} i = 0 = 0(0+1)/2$, so P(0) **is true.**
3. **Induction Hypothesis: Suppose that** P(k) **is true for some arbitrary integer** k ≥ 0**. I.e., suppose** $\sum_{i=0}^{k} i = k(k+1)/2$
4. **Induction Step: We can see that**

$$\begin{aligned}
\sum_{i=0}^{k+1} i &= \left(\sum_{i=0}^{k} i\right) + (k+1) \\
&= \mathrm{k}(\mathrm{k}+1)/2 + (k+1) && \textbf{by IH} \\
&= (\mathrm{k}+1)(\mathrm{k}/2+1) \\
&= (\mathrm{k}+1)(\mathrm{k}+2)/2
\end{aligned}$$

which is exactly P(k+1)**.**

5. **Thus** P(n) **is true for all** n ≥ 0**, by induction.**

Induction: Changing the starting point

- What if we want to prove that $P(n)$ is true for all integers $n \geq b$ for some integer b?

- Define predicate $Q(k) = P(k+b)$ for all k.
  - Then $\forall n\, Q(n) \equiv \forall n \geq b \;\; P(n)$

- Ordinary induction for Q:
  - Prove $Q(0) \equiv P(b)$
  - Prove
    $$\forall k \left(Q(k) \longrightarrow Q(k+1)\right) \equiv \forall k \geq b\left(P(k) \longrightarrow P(k+1)\right)$$

Inductive Proofs In 5 Easy Steps

Template for induction from a different base case

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

   Assume $P(k)$ is true for an arbitrary integer $k \geq \boldsymbol{b}$"
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 $n \geq \boldsymbol{b}$"

Prove $3^n \geq n^2 + 3$ for all $n \geq 2$

Prove $3^n \geq n^2 + 3$ for all $n \geq 2$

1. **Let** P(n) **be** "$3^n \geq n^2+3$". **We will show** P(n) **is true for all integers** $n \geq 2$ **by induction.**

Prove $3^n \geq n^2 + 3$ for all $n \geq 2$

1. **Let** P(n) **be** "$3^n \geq n^2+3$". **We will show** P(n) **is true for all integers** $n \geq 2$ **by induction.**
2. **Base Case** (n=2): $3^2 = 9 \geq 7 = 4+3 = 2^2+3$ **so** P(2) **is true.**

Prove $3^n \geq n^2 + 3$ for all $n \geq 2$

1. **Let** P(n) **be** "$3^n \geq n^2+3$". **We will show** P(n) **is true for all integers** $n \geq 2$ **by induction.**
2. **Base Case** (n=2): $3^2 = 9 \geq 7 = 4+3 = 2^2+3$ **so** P(2) **is true.**
3. **Inductive Hypothesis: Suppose that** P(k) **is true for some arbitrary integer** $k \geq 2$. **I.e., suppose** $3^k \geq k^2+3$.

Prove $3^n \geq n^2 + 3$ for all $n \geq 2$

1. **Let** P(n) **be** "$3^n \geq n^2+3$". **We will show** P(n) **is true for all integers** $n \geq 2$ **by induction.**
2. **Base Case** (n=2): $3^2 = 9 \geq 7 = 4+3 = 2^2+3$ **so** P(2) **is true.**
3. **Inductive Hypothesis: Suppose that** P(k) **is true for some arbitrary integer** $k \geq 2$**. I.e., suppose** $3^k \geq k^2+3$**.**
4. **Inductive Step:**

Goal: Show P(k+1), **i.e. show** $3^{k+1} \geq (k+1)^2+3=k^2+2k+4$

Prove $3^n \geq n^2 + 3$ for all $n \geq 2$

1. **Let** P(n) **be** "$3^n \geq n^2+3$". **We will show** P(n) **is true for all integers** $n \geq 2$ **by induction.**
2. **Base Case** (n=2): $3^2 = 9 \geq 7 = 4+3 = 2^2+3$ **so** P(2) **is true.**
3. **Inductive Hypothesis: Suppose that** P(k) **is true for some arbitrary integer** $k \geq 2$. **I.e., suppose** $3^k \geq k^2+3$.
4. **Inductive Step: We can see that**

$$
\begin{aligned}
3^{k+1} &= 3(3^k) \\
&\geq 3(k^2+3) && \textbf{by the IH} \\
&= k^2+2k^2+9 \\
&\geq k^2+2k+9 && \textbf{since } k^2 \geq k \\
&\geq k^2+2k+4 && \textbf{since } 9 \geq 4 \\
&= (k+1)^2+3
\end{aligned}
$$

Therefore P(k+1) **is true.**

Prove $3^n \geq n^2 + 3$ for all $n \geq 2$

1. Let P(n) be "$3^n \geq n^2+3$". We will show P(n) is true for all integers $n \geq 2$ by induction.
2. Base Case (n=2): $3^2 = 9 \geq 7 = 4+3 = 2^2+3$ so P(2) is true.
3. Inductive Hypothesis: Suppose that P(k) is true for some arbitrary integer $k \geq 2$. I.e., suppose $3^k \geq k^2+3$.
4. Inductive Step: We can see that

$$
\begin{aligned}
3^{k+1} = 3(3^k) &\geq 3(k^2+3) && \text{by the IH} \\
&= k^2+2k^2+9 \\
&\geq k^2+2k+9 && \text{since } k^2 \geq k \\
&\geq k^2+2k+4 && \text{since } 9 \geq 4 \\
&= (k+1)^2+3
\end{aligned}
$$

   Therefore P(k+1) is true.

5. Thus P(n) is true for all integers $n \geq 2$, by induction.

Induction: Adding Base Cases

- **What if we want to prove that $P(n)$ is true for all integers $n \geq b$ for some integer b but the inductive step only works for $n \geq c$?**

- **Add proofs of $P(b)$, $P(b+1)$, ..., $P(c-1)$**
  - **will call these extra "base cases"**

- **Formally, we are using the fact that**

$$P(b) \land \cdots \land P(c-1) \land \forall n\,((c \leq n) \to P(n))$$
$$\equiv \forall n\,((b \leq n) \to P(n))$$

Inductive Proofs In 5 Easy Steps

Template for induction with multiple base cases

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

   Assume $P(k)$ is true for an arbitrary integer $k \geq \boldsymbol{c}$"
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 $n \geq \boldsymbol{b}$"

Recursive Definitions of Functions

Familiar Recursive Definitions

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

Then we have familiar summation notation:

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

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

There is also product notation:

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

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

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 "$n! \leq n^n$". We will show that $P(n)$ is true for all integers $n \geq 1$ by induction.

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.**

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 $k! \leq k^k$.

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** $k! \leq k^k$.
4. **Inductive Step:**

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

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:**

   We can calculate:

$$
\begin{aligned}
(k+1)! &= (k+1) \cdot k! && \text{by definition of !} \\
&\leq (k+1) \cdot k^k && \text{by the IH} \\
&\leq (k+1) \cdot (k+1)^k && \text{since } k \geq 0 \\
&= (k+1)^{k+1}
\end{aligned}
$$

   Therefore P(k+1) **is true.**

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:**

   We can calculate:

$$
\begin{aligned}
(k+1)! &= (k+1)\cdot k! && \textbf{by definition of !} \\
&\leq (k+1)\cdot k^k && \textbf{by the IH} \\
&\leq (k+1)\cdot (k+1)^k && \textbf{since } k \geq 0 \\
&= (k+1)^{k+1}
\end{aligned}
$$

   Therefore P(k+1) **is true.**

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

Induction Is A Rule of Inference

Domain: Non-Negative Numbers

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

Induction

How do the givens prove P(5)?

$$P(0) \xrightarrow{P(0)\to P(1)} P(1) \xrightarrow{P(1)\to P(2)} P(2) \xrightarrow{P(2)\to P(3)} P(3) \xrightarrow{P(3)\to P(4)} P(4) \xrightarrow{P(4)\to P(5)} P(5)$$

Induction Is A Rule of Inference

Domain: Non-Negative Numbers

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

How do the givens prove P(5)?

$$P(0) \xrightarrow{P(0)\to P(1)} P(1) \xrightarrow{P(1)\to P(2)} P(2) \xrightarrow{P(2)\to P(3)} P(3) \xrightarrow{P(3)\to P(4)} P(4) \xrightarrow{P(4)\to P(5)} P(5)$$

First, we have P(0).
Since P(n) → P(n+1) for all n, we have P(0) → P(1).
Since P(0) is true and P(0) → P(1), by Modus Ponens, P(1) is true.
Since P(n) → P(n+1) for all n, we have P(1) → P(2).
Since P(1) is true and P(1) → P(2), by Modus Ponens, P(2) is true.

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

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 $\forall n\, Q(n) \equiv \forall n\, P(n)$

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:

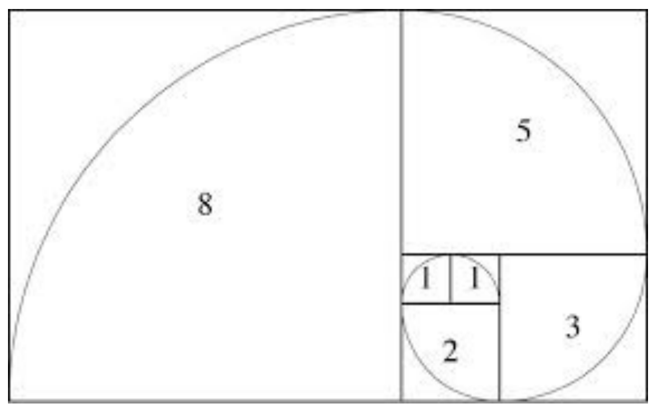
   Use the goal to figure out what you need.

   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 $n \geq b$"

Fibonacci Numbers

$$f_0 := 0$$
$$f_1 := 1$$
$$f_{n+2} := f_{n+1} + f_n$$

Will need facts about f_{n-2} to reason about f_n



Bounding Fibonacci: $f_n < 2^n$ 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.**

$$f_0 = 0 \qquad f_1 = 1$$
$$f_{n+2} = f_{n+1} + f_n$$

Bounding Fibonacci: $f_n < 2^n$ 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.**

Bounding Fibonacci: $f_n < 2^n$ 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**.**

$$\boldsymbol{f_0 = 0 \quad f_1 = 1}$$
$$\boldsymbol{f_{n+2} = f_{n+1} + f_n}$$

Bounding Fibonacci: $f_n < 2^n$ 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:

$f_{k+1} = f_k + f_{k-1}$ def of f

Oops! This is only true if $k + 1 \geq 2$!

Goal: Show P(k+1); that is, $f_{k+1} < 2^{k+1}$

$f_0 = 0 \quad f_1 = 1$
$f_{n+2} = f_{n+1} + f_n$

Bounding Fibonacci: $f_n < 2^n$ 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 and**
   $f_1 = 1 < 2 = 2^1$ **so** P(1) **is true.**

$$f_0 = 0 \quad f_1 = 1$$
$$f_{n+2} = f_{n+1} + f_n$$

Bounding Fibonacci: $f_n < 2^n$ 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 and**
   $f_1 = 1 < 2 = 2^1$ **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**.**

$$\boldsymbol{f_0 = 0 \quad f_1 = 1}$$
$$\boldsymbol{f_{n+2} = f_{n+1} + f_n}$$

Bounding Fibonacci: $f_n < 2^n$ 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 and**
   $f_1 = 1 < 2 = 2^1$ **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,** $f_{k+1} < 2^{k+1}$

$$f_0 = 0 \quad f_1 = 1$$
$$f_{n+2} = f_{n+1} + f_n$$

Bounding Fibonacci: $f_n < 2^n$ 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 and**
   $f_1 = 1 < 2 = 2^1$ **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: We can calculate that**

$$
\begin{aligned}
f_{k+1} &= f_k + f_{k-1} && \textbf{def of f (since } k+1 \geq 2\textbf{)} \\
&< 2^k + 2^{k-1} && \textbf{by IH (since } k-1 \geq 0\textbf{)} \\
&< 2^k + 2^k \\
&= 2 \cdot 2^k \\
&= 2^{k+1}
\end{aligned}
$$

so P(k+1) **is true.**

$$\boxed{\begin{array}{l} f_0 = 0 \quad f_1 = 1 \\ f_{n+2} = f_{n+1} + f_n \end{array}}$$

Bounding Fibonacci: $f_n < 2^n$ 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 and**
   $f_1 = 1 < 2 = 2^1$ **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: We can calculate that**

$$
\begin{aligned}
f_{k+1} &= f_k + f_{k-1} && \textbf{def of f (since } k+1 \geq 2) \\
&< 2^k + 2^{k-1} && \textbf{by IH (since } k-1 \geq 0) \\
&< 2^k + 2^k \\
&= 2^{k+1}
\end{aligned}
$$

   so P(k+1) **is true.**

5. **Therefore, by strong induction,** $f_n < 2^n$ **for all integers** $n \geq 0$.

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.

$$f_0 = 0 \quad f_1 = 1$$
$$f_{n+2} = f_{n+1} + f_n$$

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 Case: $f_2 = f_1 + f_0 = 1 \geq 1 = 2^0 = 2^{2/2-1}$ so P(2) holds

$$\boldsymbol{f_0 = 0 \quad f_1 = 1}$$
$$\boldsymbol{f_{n+2} = f_{n+1} + f_n}$$

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 Case: $f_2 = f_1 + f_0 = 1 \geq 1 = 2^0 = 2^{2/2-1}$ so P(2) holds
3. Inductive Hypothesis: Assume that for some arbitrary integer $k \geq 2$, P(j) is true for every integer j from 2 to k.

$$\boldsymbol{f_0 = 0 \quad f_1 = 1}$$
$$\boldsymbol{f_{n+2} = f_{n+1} + f_n}$$

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 Case: $f_2 = f_1 + f_0 = 1 \geq 1 = 2^0 = 2^{2/2 - 1}$ so P(2) holds
3. Inductive Hypothesis: Assume that for some arbitrary integer $k \geq 2$, P(j) is true for every integer j from 2 to k.
4. Inductive Step:

Goal: Show P(k+1); that is, $f_{k+1} \geq 2^{(k+1)/2 -1}$

$$\boldsymbol{f_0 = 0 \quad f_1 = 1}$$
$$\boldsymbol{f_{n+2} = f_{n+1} + f_n}$$

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 Case:** $f_2 = f_1 + f_0 = 1 \geq 1 = 2^0 = 2^{2/2-1}$ **so** P(2) **holds**
3. **Inductive Hypothesis: Assume that for some arbitrary integer** $k \geq 2$, P(j) **is true for every integer** j **from** 2 **to** k.
4. **Inductive Step:**

$$
\begin{aligned}
f_{k+1} &= f_k + f_{k-1} && \textbf{def of f (since } k+1 \geq 2\textbf{)} \\
&\geq 2^{k/2-1} + f_{k-1} && \textbf{by the IH} \\
&\geq 2^{k/2-1} + 2^{(k-1)/2-1} && \textbf{by the IH}
\end{aligned}
$$

Oops! This is only true if $k - 1 \geq 2$ **!**

Goal: Show P(k+1)**; that is,** $f_{k+1} \geq 2^{(k+1)/2-1}$

$f_0 = 0 \quad f_1 = 1$

$f_{n+2} = f_{n+1} + f_n$

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:** $f_2 = f_1 + f_0 = 1 \geq 1 = 2^0 = 2^{2/2 - 1}$ **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,** $f_{k+1} \geq 2^{(k+1)/2 - 1}$

$f_0 = 0 \quad f_1 = 1$
$f_{n+2} = f_{n+1} + f_n$

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:** $f_2 = f_1 + f_0 = 1 \geq 1 = 2^0 = 2^{2/2-1}$ **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: We can calculate**

$f_{k+1} = f_k + f_{k-1}$	**def of f (since** $k+1 \geq 4$**)**
$\geq 2^{k/2-1} + f_{k-1}$	**by the IH**
$\geq 2^{k/2-1} + 2^{(k-1)/2-1}$	**by the IH (since** $k-1 \geq 2$**)**
$\geq 2 \cdot 2^{(k-1)/2-1}$	
$= 2^{(k+1)/2-1}$	

so P(k+1) **is true.**

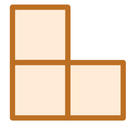
Bounding Fibonacci II: $f_n \geq 2^{n/2-1}$ for all $n \geq 2$

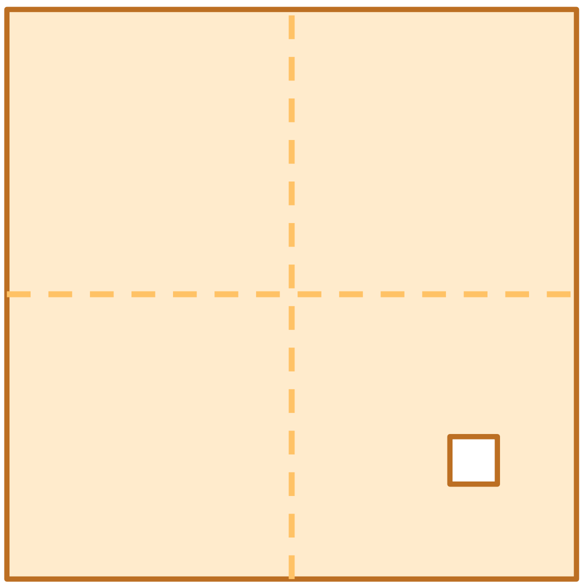
1. Let $P(n)$ be "$f_n \geq 2^{n/2-1}$". We prove that $P(n)$ is true for all integers $n \geq 2$ by strong induction.
2. Base Cases: $f_2 = f_1 + f_0 = 1 \geq 1 = 2^0 = 2^{2/2-1}$ so $P(2)$ holds
   $f_3 = f_2 + f_1 = 2 \geq 2^{1/2} = 2^{3/2-1}$ so $P(3)$ holds
3. Inductive Hypothesis: Assume that for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

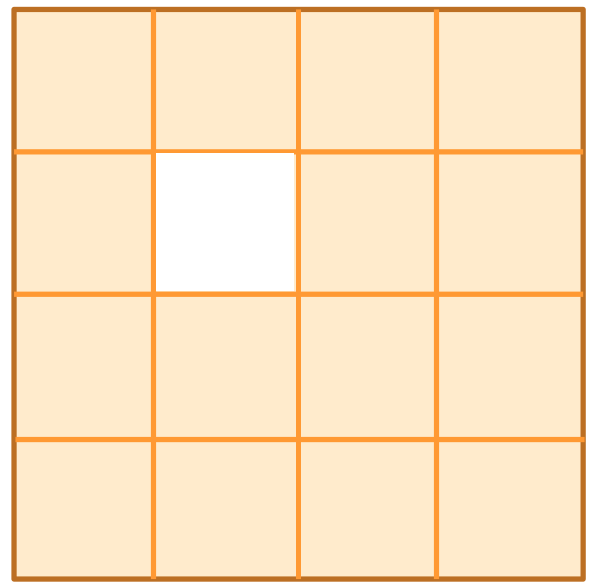
$$
\begin{aligned}
f_{k+1} &= f_k + f_{k-1} && \text{def of f (since } k+1 \geq 4\text{)} \\
&\geq 2^{k/2-1} + f_{k-1} && \text{by the IH} \\
&\geq 2^{k/2-1} + 2^{(k-1)/2-1} && \text{by the IH (since } k-1 \geq 2\text{)} \\
&\geq 2 \cdot 2^{(k-1)/2-1} = 2^{(k+1)/2-1}
\end{aligned}
$$

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

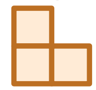
Checkerboard Tiling

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

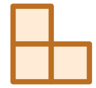
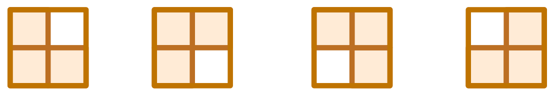




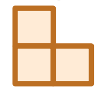
Checkerboard Tiling

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.

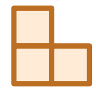
Checkerboard Tiling

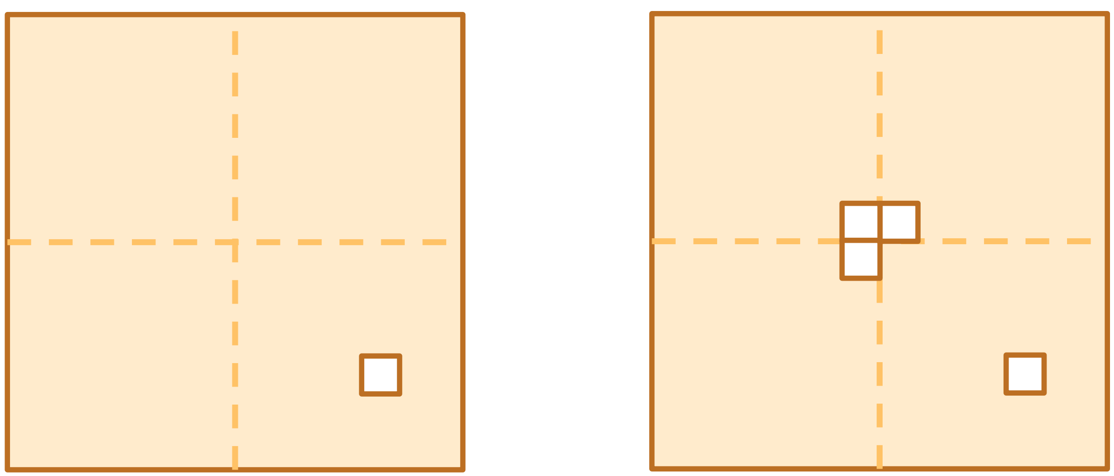
1. **Let** $P(n)$ **be any** $2^n \times 2^n$ **checkerboard with one square removed can be tiled with**  **.**
   We prove $P(n)$ **for all** $n \geq 1$ **by induction on** n.
2. **Base Case:** $n=1$ 

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.

Applications

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:

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

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

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.

Algorithmic Problems

- **Multiplication**
  - Given primes $p_1, p_2, \dots, p_k$, calculate their product $p_1 p_2 \dots p_k$
- **Factoring**
  - Given an integer n, determine the prime factorization of n

Factoring

Factor the following 232 digit number [RSA768]:

123018668453011775513049495838496272077
285356959533479219732245215172640050726
365751874520219978646938995647494277406
384592519255732630345373154826850791702
612214291346167042921431160222124047927
4737794080665351419597459856902143413

1230186684530117755130494958384962720772853569595334792197322452151726400507263657518745202199786469389956474942774063845925192557326303453731548268507917026122142913461670429214311602221240479274737794080665351419597459856902143413

=

33478071698956898786044169848212690817704794983713768568912431388982883793878002287614711652531743087737814467999489

×

36746043666799590428244633799627952632279158164343087642676032283815739666511279233373417143396810270092798736308917

Famous Algorithmic Problems

- Factoring
  - Given an integer n, determine the prime factorization of n
- Primality Testing
  - Given an integer n, determine if n is prime

- Factoring is hard
  - (on a classical computer)
- Primality Testing is easy

GCD and Factoring

$a = 2^3 \cdot 3 \cdot 5^2 \cdot 7 \cdot 11 = 46{,}200$

$b = 2 \cdot 3^2 \cdot 5^3 \cdot 7 \cdot 13 = 204{,}750$

$$\mathrm{GCD}(a, b) = 2^{\min(3,1)} \cdot 3^{\min(1,2)} \cdot 5^{\min(2,3)} \cdot 7^{\min(1,1)} \cdot 11^{\min(1,0)} \cdot 13^{\min(0,1)}$$

Factoring is hard

Yet, we can compute GCD(a,b) without factoring!

Basic Applications of mod

- Two's Complement
- Hashing
- Pseudo random number generation

n-bit Unsigned Integer Representation

- Represent integer x as sum of powers of 2:

99	= 64 + 32 + 2 + 1	= $2^6 + 2^5 + 2^1 + 2^0$
18	= 16 + 2	= $2^4 + 2^1$

- Binary representation shows which powers are used:

99: 0110 0011
18: 0001 0010

n-bit Unsigned Integer Representation

- Suppose we write numbers with 4 bits:

14	= 8 + 4 + 2	$= 2^3 + 2^2 + 2^1$	= 1110
11	= 8 + 2 + 1	$= 2^3 + 2^1 + 2^0$	= 1011

- Largest number we can write in 4 bits is:

15	= 8 + 4 + 2 + 1	$= 2^3 + 2^2 + 2^1 + 2^0$	= 1111

- Note that $15 = 16 – 1 = 2^4 - 1$
  - we proved this before!

n-bit Unsigned Integer Representation

- Suppose we write numbers with 4 bits (0 .. 15):

14	= 8 + 4 + 2	= $2^3 + 2^2 + 2^1$	= 1110
11	= 8 + 2 + 1	= $2^3 + 2^1 + 2^0$	= 1011

- Adding these numbers gives us 25 with 5 bits:

25	= 16 + 8 + 1	= $2^4 + 2^3 + 2^0$	= 11001

- If we drop the highest bit, we have

9	= 8 + 1	= $2^3 + 2^0$	= 1001

n-bit Unsigned Integer Representation

25	= 16 + 8 + 1	$= 2^4 + 2^3 + 2^0$	= 11001
9	= 8 + 1	$= 2^3 + 2^0$	= 1001

- Note that $9 \equiv_{16} 25$ since $25 - 9 = 16$
  - dropping 2^4 bit subtracts 16
  - dropping 2^5 bit subtracts $32 = 2 \cdot 16$
  - dropping 2^6 bit subtracts $64 = 4 \cdot 16$
- Throwing away all but 4 bits is arithmetic mod 16
  - easier to implement normal arithmetic!

Sign-Magnitude Integer Representation

n-bit signed integers
Suppose that $-2^{n-1} < x < 2^{n-1}$
First bit as the sign, $n-1$ bits for the value

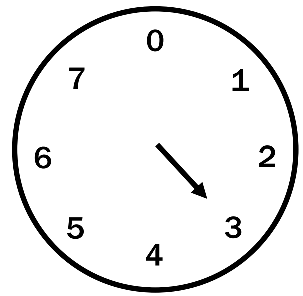
99 = 64 + 32 + 2 + 1
18 = 16 + 2

For n = 8:
99: 0110 0011
-18: 1001 0010

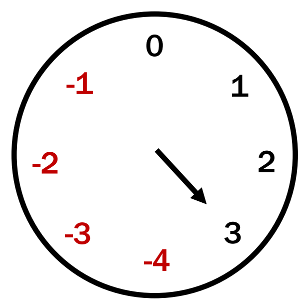
Problem: this has both +0 and -0 (annoying)

Arithmetic on a Clock

3 bits, unsigned



3 bits, signed



Since $-1 \equiv_8 7$, arithmetic is unchanged

Only differences are printing and comparison

Two's Complement Representation

Suppose that $0 \leq x < 2^{n-1}$

x is represented by the binary representation of x

Suppose that $-2^{n-1} \leq x < 0$

x is represented by the binary representation of $x + 2^n$

result is in the range $2^{n-1} \leq x < 2^n$

99 = 64 + 32 + 2 + 1

18 = 16 + 2

For n = 8:

99: 0110 0011

-18: 1110 1110 (-18 + 256 = 238)

Two's Complement Representation

Suppose that $0 \leq x < 2^{n-1}$

x is represented by the binary representation of x

Suppose that $-2^{n-1} \leq x < 0$

x is represented by the binary representation of $x + 2^n$

result is in the range $2^{n-1} \leq x < 2^n$

With 4 bits:

0	1	2	3	4	5	6	7	-8	-7	-6	-5	-4	-3	-2	-1
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

Key property: First bit is still the sign bit!

Key property: Twos complement representation of any number $\boldsymbol{y}$ is equivalent to $\boldsymbol{y} \bmod \mathbf{2}^{\boldsymbol{n}}$ so arithmetic works $\bmod \mathbf{2}^{\boldsymbol{n}}$

$$y + 2^n \equiv_{2^n} y$$

I’m ALIVE!

```
public class Test {
    final static int SEC_IN_YEAR = 365*24*60*60;
    public static void main(String args[]) {
        System.out.println(
            “I will be alive for at least ” +
            SEC_IN_YEAR * 101 + “ seconds.”
        );
    }
}
```

```
 ----jGRASP exec: java Test
I will be alive for at least -186619904 seconds.

 ----jGRASP: operation complete.
```

Two's Complement Representation

- For $0 < x \leq 2^{n-1}$, $-x$ is represented by the binary representation of $-x + 2^n$
  - How do we calculate –x from x?
  - E.g., what happens for "`return -x;`" in Java?

$$-x + 2^n = (2^n - 1) - \mathrm{x} + 1$$

- To compute this, flip the bits of x then add 1!

  Flip the bits of x means replace x by $2^n - 1 - x$

  Then add 1 to get $-x + 2^n$