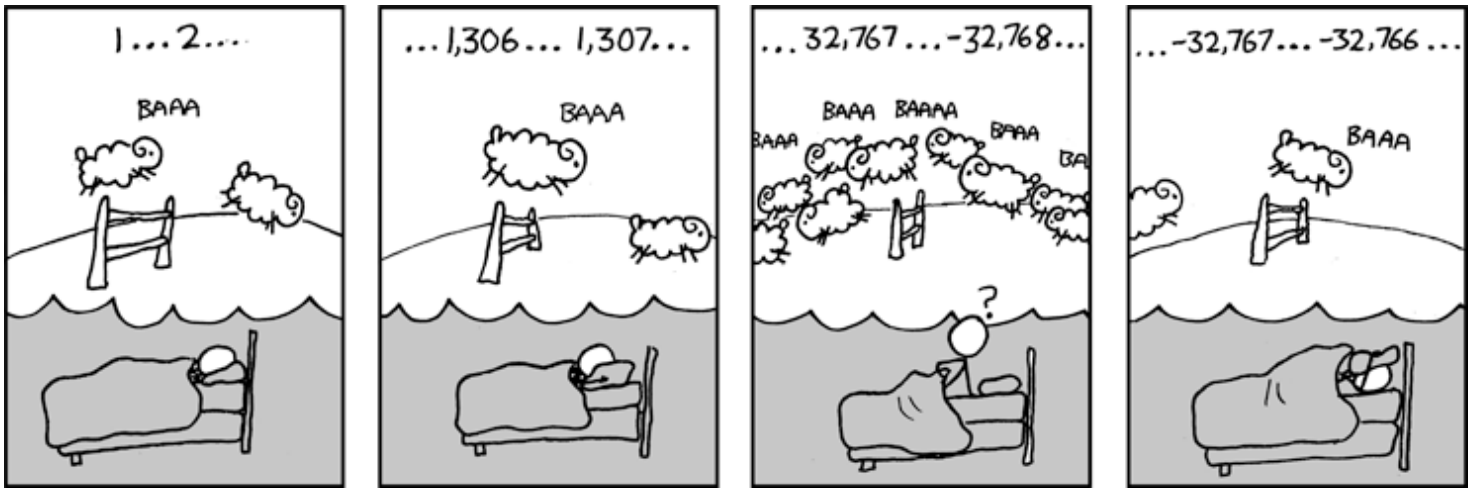
CSE 311: Foundations of Computing

Topic 5: More Number Theory



GCD

Greatest Common Divisor

GCD(a, b):

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

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

d is GCD iff $(d \mid a) \wedge (d \mid b) \wedge \forall x\, (((x \mid a) \wedge (x \mid b)) \rightarrow (x \leq d))$

Useful GCD Fact

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

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 $d \mid a$.

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, $d \mid (a \bmod b)$ also.

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

Another simple GCD fact

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

Euclid’s Algorithm

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

```
int gcd(int a, int b){ /* Assumes: a >= b, 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 $\gcd(g, 0) = g$.

gcd(660,126) = gcd(126, 660 mod 126) = gcd(126, 30)
= gcd(30, 126 mod 30) = gcd(30, 6)
= gcd(6, 30 mod 6) = gcd(6, 0)
= 6

Bézout's theorem

If a and b are positive integers, then there exist integers $\boldsymbol{s}$ and $\boldsymbol{t}$ 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):

a b b a mod b = r b r

$$\gcd(35, 27) = \gcd(27, 35 \bmod 27) = \gcd(27, 8)$$

a = q * b + r

$$35 = 1 * 27 + 8$$

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):

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

$$= \gcd(8, 27 \bmod 8) \quad = \gcd(8, 3)$$

$$= \gcd(3, 8 \bmod 3) \quad = \gcd(3, 2)$$

$$= \gcd(2, 3 \bmod 2) \quad = \gcd(2, 1)$$

$$= \gcd(1, 2 \bmod 1) \quad = \gcd(1, 0)$$

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

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

$$35 = 1 * 27 + 8$$
$$27 = 3 * 8 + 3$$
$$8 = 2 * 3 + 2$$
$$3 = 1 * 2 + 1$$

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

$$35 = 1 * 27 + 8$$
$$27 = 3 * 8 + 3$$
$$8 = 2 * 3 + 2$$
$$3 = 1 * 2 + 1$$

r = a – q * b

$$8 = 35 - 1 * 27$$
$$3 = 27 - 3 * 8$$
$$2 = 8 - 2 * 3$$
$$1 = 3 - 1 * 2$$

Extended Euclidean algorithm

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

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

Step 3 (Backward Substitute Equations):

$8 = 35 - 1 * 27$

$3 = 27 - 3 * 8$

$2 = 8 - 2 * 3$

$1 = 3 - 1 * 2$

Extended Euclidean algorithm

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

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

Step 3 (Backward Substitute Equations):

$8 = 35 - 1 * 27$

$3 = 27 - 3 * 8$

$2 = 8 - 2 * 3$

$1 = 3 - 1 * 2$

Plug in the def of 2

$$1 = 3 - 1 * (8 - 2 * 3)$$
$$= 3 - 8 + 2 * 3$$
$$= (-1) * 8 + 3 * 3$$

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):

$$8 = 35 - 1 * 27$$

$$3 = 27 - 3 * 8$$

$$2 = 8 - 2 * 3$$

$$1 = 3 - 1 * 2$$

Plug in the def of 2

$$1 = 3 - 1 * (8 - 2 * 3)$$
$$= 3 - 8 + 2 * 3$$
$$= (-1) * 8 + 3 * 3$$

Re-arrange into 3's and 8's

Plug in the def of 3

$$= (-1) * 8 + 3 * (27 - 3 * 8)$$
$$= (-1) * 8 + 3 * 27 + (-9) * 8$$
$$= 3 * 27 + (-10) * 8$$

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):

$$8 = 35 - 1 * 27$$

$$3 = 27 - 3 * 8$$

$$2 = 8 - 2 * 3$$

$$1 = 3 - 1 * 2$$

Plug in the def of 2

$$1 = 3 - 1 * (8 - 2 * 3)$$
$$= 3 - 8 + 2 * 3$$
$$= (-1) * 8 + 3 * 3$$

Re-arrange into 3's and 8's

Plug in the def of 3

$$= (-1) * 8 + 3 * (27 - 3 * 8)$$
$$= (-1) * 8 + 3 * 27 + (-9) * 8$$
$$= 3 * 27 + (-10) * 8$$

Re-arrange into 8's and 27's

$$= 3 * 27 + (-10) * (35 - 1 * 27)$$
$$= 3 * 27 + (-10) * 35 + 10 * 27$$
$$= 13 * 27 + (-10) * 35$$

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 $\text{mod } m$

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…

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**

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

$$26 = 3*7 + 5 \qquad 5 = 26 - 3*7$$
$$7 = 1*5 + 2 \qquad 2 = 7 - 1*5$$
$$5 = 2*2 + 1 \qquad 1 = 5 - 2*2$$

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

Example: Solve a Modular Equation

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

Find multiplicative inverse of 7 modulo 26

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

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

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

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

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

Now $(-11) \bmod 26 = 15$.

"<u>the</u>" multiplicative inverse

(-11 is also "a" multiplicative inverse)

Example: Solve a Modular Equation

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

Find multiplicative inverse of 7 modulo 26… it's 15.

Multiplying both sides by 15 gives

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

Simplify on both sides to get

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

So, <u>all</u> solutions of this congruence are numbers of the form $x = 19 + 26k$ for some $k \in \mathbb{Z}$.

Multiplicative Inverses and Algebra

Adding to both sides easily reversible:

$$x \equiv_m y$$

$-c$ $+c$

$$x + c \equiv_m y + c$$

The same is not true of multiplication…

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

$$x \equiv_m y$$

$\times d$ $\times c$

$$cx \equiv_m cy$$

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

Solving Modular Equations

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

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** $k \in \mathbb{Z}$

(must be of the form $a + mk$ **for all** $k \in \mathbb{Z}$ **with** $0 \leq a < m$**)**

Examples Not in "Standard Form"

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

Modular equation like $\mathrm{A}x \equiv_{26} B$ for some A and B is in "standard form".

- solve by multiplying both sides by inverse of A

What about equation not in standard form?

Examples Not in "Standard Form"

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

Transform into standard form by adding to both sides

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

$7(x-3) + 21 \equiv_{26} 8 + 2x + 21$ **add** 21 **to both sides**

$7x \equiv_{26} 3 + 2x$ **simplify**

$7x - 2x \equiv_{26} 3 + 2x - 2x$ **add** $-2x$ **to both sides**

$5x \equiv_{26} 3$ **simplify**

Induction

Mathematical Induction

Method for proving statements about all natural numbers

- **A new logical inference rule!**
  - It only applies over the natural numbers
  - The idea is to **use** the special structure of the naturals to prove things more easily
- **Particularly useful for reasoning about programs!**

```
for (int i=0; i < n; n++) { … }
```

  - Show P(i) holds after i times through the loop

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 the natural numbers!

Domain: Natural Numbers

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

Induction Is A Rule of Inference

Domain: Natural Numbers

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

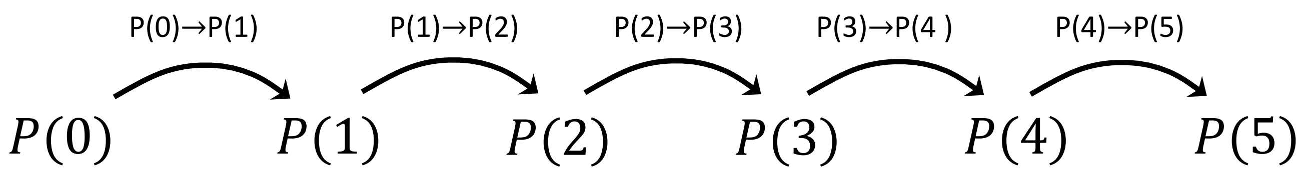
How do the givens prove P(3)?

Induction Is A Rule of Inference

Domain: Natural Numbers

$$P(0)$$
$$\forall k\ (P(k) \longrightarrow P(k+1))$$
$$\therefore \forall n\ \ P(n)$$

How do the givens prove 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

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

Using The Induction Rule In A Formal Proof

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

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

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

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

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

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

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

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 $P(k) \to P(k+1)$	Direct Proof
2. $\forall k\,(P(k) \to P(k+1))$	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) → P(k+1)	Direct Proof	**Conclusion**
2. ∀k (P(k) → P(k+1))	Intro ∀	
3. ∀n P(n)	Induction: 1, 2	

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

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

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

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 natural 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 natural 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:**

   $2^0 + 2^1 + \ldots + 2^k = 2^{k+1} - 1$ **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** $P(k+1)$**.**

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 natural 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

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

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

Alternative way of writing the inductive step

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 natural 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

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

   which is exactly P(k+1).

5. Thus P(n) is true for all $n \geq 0$, by induction.

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

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

Summation Notation

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

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

1. **Let** P(n) **be** "0 + 1 + 2 + … + n = n(n+1)/2". **We will show** P(n) **is true for all natural numbers by induction.**

Summation Notation

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

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

1. **Let** P(n) **be** "0 + 1 + 2 + … + n = n(n+1)/2". **We will show** P(n) **is true for all natural numbers by induction.**
2. **Base Case** (n=0): 0 = 0(0+1)/2. **Therefore** P(0) **is true.**

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

1. **Let** P(n) **be** "0 + 1 + 2 + … + n = n(n+1)/2". **We will show** P(n) **is true for all natural numbers by induction.**
2. **Base Case** (n=0): 0 = 0(0+1)/2. **Therefore** P(0) **is true.**
3. **Induction Hypothesis: Suppose that** P(k) **is true for some arbitrary integer** k ≥ 0**. I.e., suppose** 1 + 2 + …+ k = k(k+1)/2

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

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

1. **Let** P(n) **be** "0 + 1 + 2 + ... + n = n(n+1)/2". **We will show** P(n) **is true for all natural numbers by induction.**
2. **Base Case** (n=0): 0 = 0(0+1)/2. **Therefore** P(0) **is true.**
3. **Induction Hypothesis: Suppose that** P(k) **is true for some arbitrary integer** k ≥ 0**. I.e., suppose** 1 + 2 + ...+ k = k(k+1)/2
4. **Induction Step:**

   Goal: Show P(k+1), **i.e. show** 1 + 2 + ...+ k+ (k+1) = (k+1)(k+2)/2

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

1. **Let** P(n) **be** "0 + 1 + 2 + … + n = n(n+1)/2". **We will show** P(n) **is true for all natural numbers by induction.**
2. **Base Case** (n=0): 0 = 0(0+1)/2. **Therefore** P(0) **is true.**
3. **Induction Hypothesis: Suppose that** P(k) **is true for some arbitrary integer** $k \geq 0$**. I.e., suppose** 1 + 2 + …+ k = k(k+1)/2
4. **Induction Step:**

$$
\begin{aligned}
1 + 2 + \ldots + k + (k+1) &= (1 + 2 + \ldots + k) + (k+1) \\
&= k(k+1)/2 + (k+1) \quad \textbf{by IH} \\
&= (k+1)(k/2 + 1) \\
&= (k+1)(k+2)/2
\end{aligned}
$$

So, we have shown 1 + 2 + … + k + (k+1) = (k+1)(k+2)/2, **which is exactly** P(k+1).

5. **Thus** P(n) **is true for all** $n \in \mathbb{N}$**, by induction.**

Induction: Changing the start line

- 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
    $\forall k\,\big(Q(k) \longrightarrow Q(k+1)\big) \equiv \forall k \geq b\big(P(k) \longrightarrow P(k+1)\big)$

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** $3^{k+1} \geq (k+1)^2+3$

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

$$3^{k+1} = 3(3^k)$$

$$\geq 3(k^2+3) \text{ by the IH}$$

$$= 3k^2+9$$

$$= k^2+2k^2+9$$

$$\geq k^2+2k+4 = (k+1)^2+3 \text{ since } k \geq 1.$$

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:**

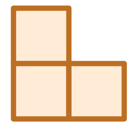
   We can see that

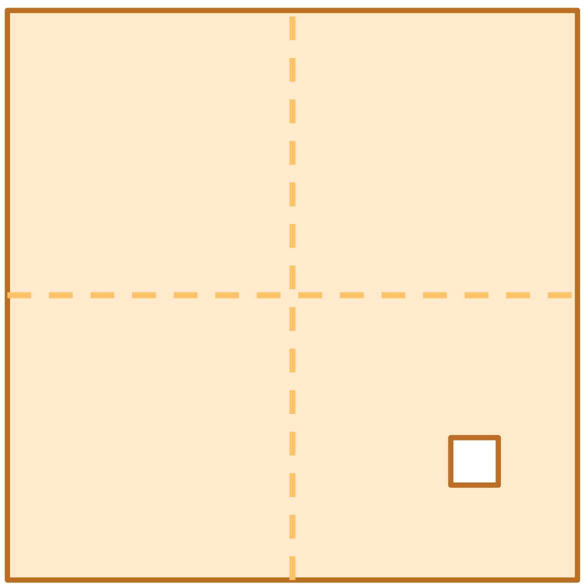
   $$3^{k+1} = 3(3^k)$$
   $$\geq 3(k^2+3) \text{ by the IH}$$
   $$= k^2+2k^2+9$$
   $$\geq k^2+2k+4 = (k+1)^2+3 \text{ since } k \geq 1.$$

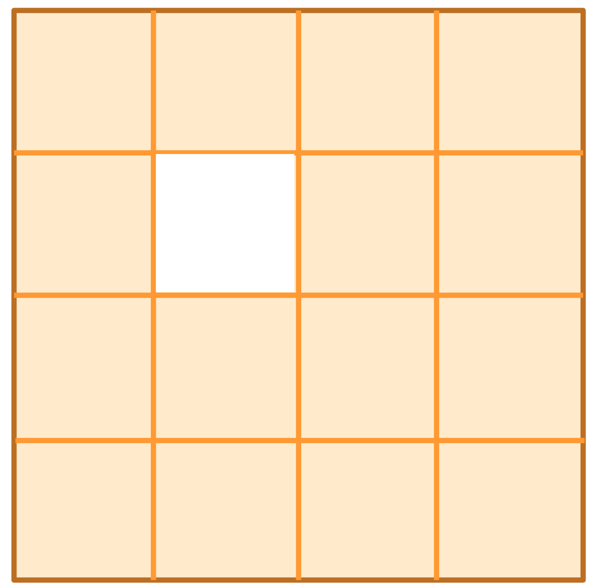
   Therefore P(k+1) **is true.**

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

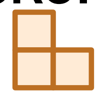
Checkerboard Tiling

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

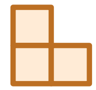




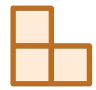
Checkerboard Tiling

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

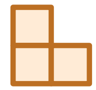
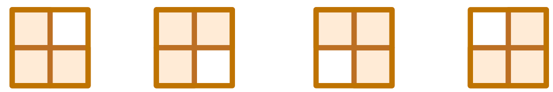
Checkerboard Tiling

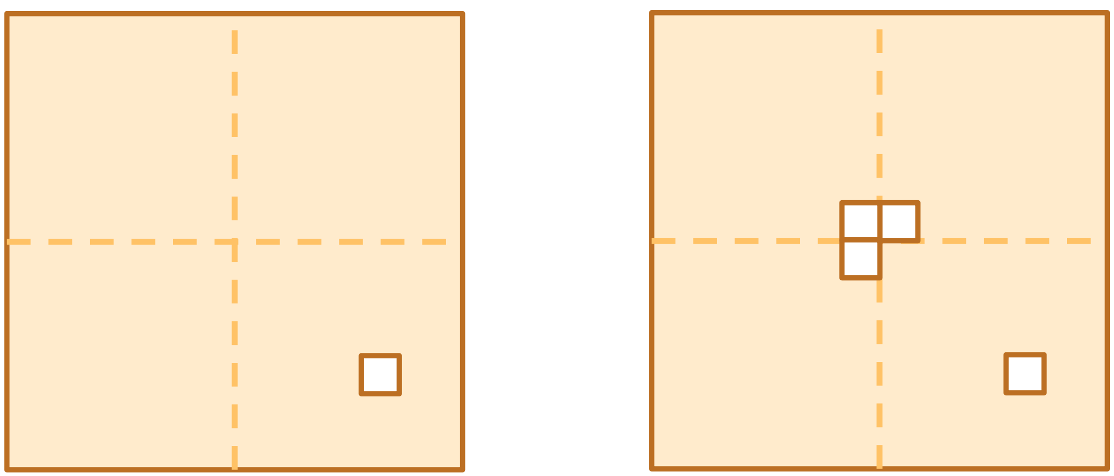
1. **Let** $P(n)$ **be any** $2^n \times 2^n$ **checkerboard with one square removed can be tiled with**  **.**
   We prove $P(n)$ **for all** $n \geq 1$ **by induction on** n.
2. **Base Case:** $n=1$ 

Checkerboard Tiling

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

Checkerboard Tiling

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



Apply IH to each quadrant then fill with extra tile.

Recall: Induction Rule of Inference

Domain: Natural Numbers

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

How do the givens prove P(5)?

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

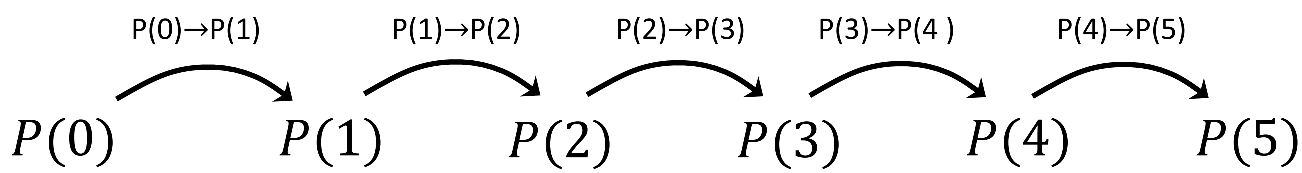
$P(0)$ $P(1)$ $P(2)$ $P(3)$ $P(4)$ $P(5)$

Recall: Induction Rule of Inference

Domain: Natural Numbers

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

How do the givens prove P(5)?



We made it harder than we needed to …

When we proved $P(2)$ **we knew BOTH** $P(0)$ **and** $P(1)$

When we proved $P(3)$ **we knew** $P(0)$ **and** $P(1)$ **and** $P(2)$

When we proved $P(4)$ **we knew** $P(0), P(1), P(2), P(3)$

etc.

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

Strong Induction

$$\frac{P(0) \qquad \forall k \left(\forall j \left(0 \leq j \leq k \rightarrow P(j) \right) \rightarrow P(k+1) \right)}{\therefore \forall n \, P(n)}$$

Strong Induction

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

Strong induction for P follows from ordinary induction for Q where

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

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

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 $n \geq b$ by induction."
2. "Base Case:" Prove $P(b)$
3. "Inductive Hypothesis:

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

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

   Use the goal to figure out what you need.

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

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

Strong Inductive Proofs In 5 Easy Steps

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

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

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

   Use the goal to figure out what you need.

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

Primality

An integer *p* greater than 1 is called *prime* if the only positive factors of *p* are 1 and *p*.

$$p > 1 \land \forall x\, ((x \mid p) \rightarrow ((x = 1) \lor (x = p)))$$

A positive integer that is greater than 1 and is not prime is called *composite*.

$$p > 1 \land \exists x\, ((x \mid p) \land (x \neq 1) \land (x \neq p))$$

Fundamental Theorem of Arithmetic

Every integer > 1 has a unique prime factorization

$48 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 3$
$591 = 3 \cdot 197$
$45{,}523 = 45{,}523$
$321{,}950 = 2 \cdot 5 \cdot 5 \cdot 47 \cdot 137$
$1{,}234{,}567{,}890 = 2 \cdot 3 \cdot 3 \cdot 5 \cdot 3{,}607 \cdot 3{,}803$

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

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

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

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

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

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

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

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

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

---

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

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

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

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

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

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

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

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

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

   Case: k+1 is prime: Then by definition k+1 **is a product of primes**
   Case: k+1 is composite: Then k+1=ab **for some integers** a **and** b **where** 2 ≤ a, b ≤ k.

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

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

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

   <u>Case: k+1 is prime</u>: Then by definition k+1 **is a product of primes**
   <u>Case: k+1 is composite:</u> Then k+1=ab **for some integers** a **and** b
   where $2 \leq a, b \leq k$. **By our IH,** P(a) **and** P(b) **are true so we have**
   $a = p_1 p_2 \cdots p_r$ **and** $b = q_1 q_2 \cdots q_s$
   for some primes $p_1, p_2, \ldots, p_r, q_1, q_2, \ldots, q_s$.
   Thus, $k+1 = ab = p_1 p_2 \cdots p_r q_1 q_2 \cdots q_s$ **which is a product of primes.**
   Since $k \geq 2$, **one of these cases must happen and so** P(k+1) **is true.**

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

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

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

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

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

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

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

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

Applications

Algorithmic Problems

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

Factoring

Factor the following 232 digit number [RSA768]:

1230186684530117755130494958384962720772
8535695953347921973224521517264005072
6365751874520219978646938995647494277406
3845925192557326303453731548268507917026
1221429134616704292143116022212404792
74737794080665351419597459856902143413

12301866845301177551304949583849627207728535695953347
92197322452151726400507263657518745202199786469389956
47494277406384592519255732630345373154826850791702612
21429134616704292143116022212404792747377940806653514
19597459856902143413

$$=$$

334780716989568987860441698482126908177047949837
137685689124313889828837938780022876147116525317
43087737814467999489

$$\times$$

367460436667995904282446337996279526322791581643
430876426760322838157396665112792333734171433968
10270092798736308917

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

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

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

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

Factoring is hard

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

Will shortly see **another** operation that
can be implemented surprisingly quickly…

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!

n-bit Unsigned Integer Representation

- Largest representable number is $2^n - 1$

$$2^n = 100...000 \qquad \text{(n+1 bits)}$$
$$2^n - 1 = 11...111 \qquad \text{(n bits)}$$

THE WALL STREET JOURNAL.

32 bits
1 = $0.0001
$429,496.7295 max

Berkshire Hathaway Inc. (BRK-A)
NYSE - Nasdaq Real Time Price. Currency in USD

436,401.00 +679.50 (+0.16%)
At close: 4:00PM EDT

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)

Two's Complement Representation

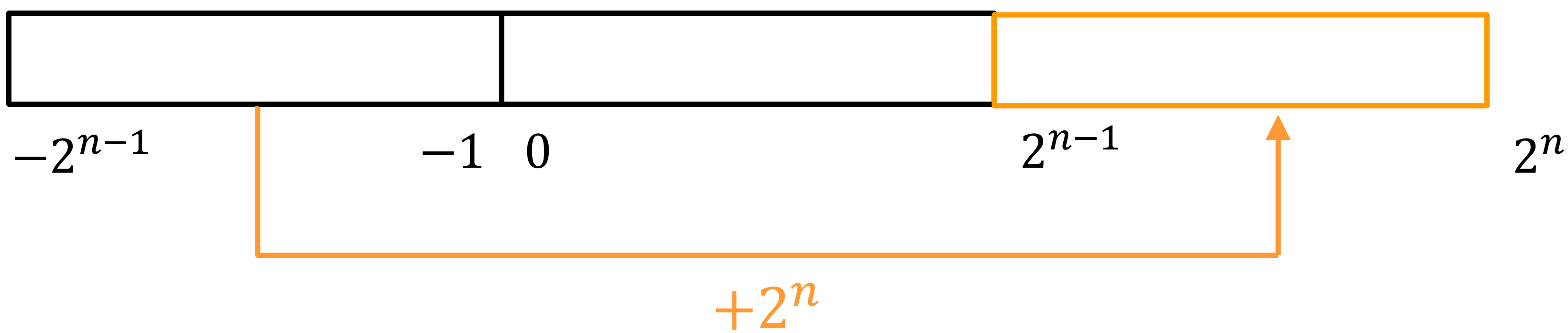
Suppose that $0 \leq x < 2^{n-1}$

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



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

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

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

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 $0 \le x < 2^{n-1}$

x is represented by the binary representation of x

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

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

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

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 $\boldsymbol{y} \bmod \mathbf{2^n}$ so arithmetic works $\bmod \mathbf{2^n}$

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

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 $-x + 2^n$

Exponentiation

- **Compute** 78365^{81453}
- **Compute** $78365^{81453} \bmod 104729$
- **Output is small**
  - need to keep intermediate results small

Small Multiplications

Since $b = qm + (b \bmod m)$, we have $b \bmod m \equiv_m b$.

And since $c = tm + (c \bmod m)$, we have $c \bmod m \equiv_m c$.

Multiplying these gives $(b \bmod m)(c \bmod m) \equiv_m bc$.

By the Lemma from a few lectures ago, this tells us
$bc \bmod m = (b \bmod m)(c \bmod m) \bmod m$.

Okay to mod b and c by m before multiplying if we are planning to mod the result by m

Repeated Squaring – small and fast

Since $b \bmod m \equiv_m b$ and $c \bmod m \equiv_m c$

we have $bc \bmod m = (b \bmod m)(c \bmod m) \bmod m$

So $a^2 \bmod m = (a \bmod m)^2 \bmod m$

and $a^4 \bmod m = (a^2 \bmod m)^2 \bmod m$

and $a^8 \bmod m = (a^4 \bmod m)^2 \bmod m$

and $a^{16} \bmod m = (a^8 \bmod m)^2 \bmod m$

and $a^{32} \bmod m = (a^{16} \bmod m)^2 \bmod m$

Can compute $a^k \bmod m$ for $k = 2^i$ in only i steps

What if k is not a power of 2?

Fast Exponentiation Algorithm

81453 in binary is 10011111000101101

$81453 = 2^{16} + 2^{13} + 2^{12} + 2^{11} + 2^{10} + 2^{9} + 2^{5} + 2^{3} + 2^{2} + 2^{0}$

$a^{81453} = a^{2^{16}} \cdot a^{2^{13}} \cdot a^{2^{12}} \cdot a^{2^{11}} \cdot a^{2^{10}} \cdot a^{2^{9}} \cdot a^{2^{5}} \cdot a^{2^{3}} \cdot a^{2^{2}} \cdot a^{2^{0}}$

a^{81453} mod m=

(…(((((a$^{2^{16}}$ mod m ·

a$^{2^{13}}$ mod m) mod m ·

a$^{2^{12}}$ mod m) mod m ·

a$^{2^{11}}$ mod m) mod m ·

a$^{2^{10}}$ mod m) mod m ·

a$^{2^{9}}$ mod m) mod m ·

a$^{2^{5}}$ mod m) mod m ·

a$^{2^{3}}$ mod m) mod m ·

a$^{2^{2}}$ mod m) mod m ·

a$^{2^{0}}$ mod m) mod m

Uses only 16 + 9 = 25 multiplications

The fast exponentiation algorithm computes $a^k \bmod m$ using $\leq 2\log k$ multiplications $\bmod m$

Using Fast Modular Exponentiation

- Your e-commerce web transactions use SSL (Secure Socket Layer) based on RSA encryption
- RSA
  - Vendor chooses random 512-bit or 1024-bit primes p, q and 512/1024-bit exponent e. Computes $m = p \cdot q$
  - Vendor broadcasts (m, e)
  - To send a to vendor, you compute $C = a^e \bmod m$ using *fast modular exponentiation* and send C to the vendor.
  - Using secret p, q the vendor computes d that is the *multiplicative inverse* of e mod $(p-1)(q-1)$.
  - Vendor computes $C^d \bmod m$ using *fast modular exponentiation*.
  - **Fact:** $a = C^d \bmod m$ for $0 < a < m$ unless $p|a$ or $q|a$

Hashing

Scenario:

Map a small number of data values from a large domain $\{0, 1, \ldots, M-1\}$ …

…into a small set of locations $\{0, 1, \ldots, n-1\}$ so one can quickly check if some value is present

- $\text{hash}(x) = x \bmod p$ for p a prime close to n
  – or $\text{hash}(x) = (ax + c) \bmod p$
- Latter depends on all the bits of the data
  – $\text{hash}(x)$ and $\text{hash}(x+1)$ can be very far apart

Hashing

- $\text{hash}(x) = (ax + c) \bmod p$ **for prime** p
  - **deterministic function with random-ish behavior**

- **Suppose that** $\text{hash}(x) = \text{hash}(y)$**...**

$\text{ax} + \text{c} \equiv_p \text{ay} + \text{c}$

$\text{ax} \equiv_p \text{ay}$ add $-\text{c}$ to both sides

$\text{x} \equiv_p \text{y}$ multiply both sides by s where $\text{as} \equiv_p 1$

- **Output as evenly spread as** $\text{hash}(x) = x \bmod p$

Hashing

- $\mathrm{hash}(x) = (ax + c) \bmod p$ **for prime p**
  - **deterministic function with random-ish behavior**

- **Applications**
  - **map integer to location in array (hash tables)**
  - **map user ID or IP address to machine**

    requests from the same user / IP address go to the same machine

    requests from different users / IP addresses spread randomly

Pseudo-Random Number Generation

Linear Congruential method

$$x_{n+1} = (a\, x_n + c) \bmod m$$

Choose random x_0**,** a**,** c**,** m **and produce a long sequence of** x_n**'s**