Warm up: Show that if a^2 if even then a is even.

Number Theory Proofs

CSE 311 Autumn 20
Lecture 14

Bézout's Theorem

Bézout's Theorem

If a and b are positive integers, then there exist integers s and t such that
gcd(a,b)$= sa + tb$

We're not going to prove this theorem…

But we'll show you how to find s,t for any positive integers a, b.

Extended Euclidian Algorithm

Step 1 compute gcd(a,b); keep tableau information.

Step 2 solve all equations for the remainder.

Step 3 substitute backward

$$\boxed{\begin{array}{l} 8 = 35 - 1 \cdot 27 \\ 3 = 27 - 3 \cdot 8 \\ 2 = 8 - 2 \cdot 3 \\ 1 = 3 - 1 \cdot 2 \end{array}}$$

gcd(27,35) = $13 \cdot 27 + (-10) \cdot 35$

$$\begin{aligned} 1 &= 3 - 1 \cdot 2 \\ &= 3 - 1 \cdot (8 - 2 \cdot 3) \\ &= -1 \cdot 8 + 3 \cdot 3 \\ &= -1 \cdot 8 + 3(27 - 3 \cdot 8) \\ &= 3 \cdot 27 - 10 \cdot 8 \\ &= 3 \cdot 27 - 10(35 - 1 \cdot 27) \\ &= 13 \cdot 27 - 10 \cdot 35 \end{aligned}$$

When substituting back, you keep the larger of m, n and the number you just substituted.
Don’t simplify further! (or you lose the form you need)

So…what's it good for?

Suppose I want to solve $7x \equiv 1 (mod\ n)$

Just multiply both sides by $\frac{1}{7}$…

Oh wait. We want a number to multiply by 7 to get 1.

If gcd(7,n) = 1

Then $s \cdot 7 + tn = 1$, so $7s - 1 = -tn$ i.e. $n | (7s - 1)$ so $7s \equiv 1 (mod\ n)$.

So the s from Bézout's Theorem is what we should multiply by!

Try it

Solve the equation $7y \equiv 3(mod\ 26)$

What do we need to find?

The multiplicative inverse of $7(\mathbf{mod}\ 26)$

Finding the inverse…

gcd(26,7) = gcd(7, 26%7) = gcd(7,5)

= gcd(5, 7%5) = gcd(5,2)

= gcd(2, 5%2) = gcd(2, 1)

= gcd(1, 2%1) = gcd(1,0)= 1.

$$1 = 5 - 2 \cdot 2$$
$$= 5 - 2(7 - 5 \cdot 1)$$
$$= 3 \cdot 5 - 2 \cdot 7$$
$$= 3 \cdot (26 - 3 \cdot 7) - 2 \cdot 7$$
$$3 \cdot 26 - 11 \cdot 7$$

-11 is a multiplicative inverse.
We'll write it as 15, since we're working mod 26.

$26 = 3 \cdot 7 + 5\ ;\ 5 = 26 - 3 \cdot 7$

$7 = 5 \cdot 1 + 2\ ;\ 2 = 7 - 5 \cdot 1$

$5 = 2 \cdot 2 + 1\ ;\ 1 = 5 - 2 \cdot 2$

Try it

Solve the equation $7y \equiv 3(mod\ 26)$

What do we need to find?
The multiplicative inverse of 7 $(mod\ 26)$.

$15 \cdot 7 \cdot y \equiv 15 \cdot 3(mod\ 26)$

$y \equiv 45(mod\ 26)$

Or $y \equiv 19(mod\ 26)$

So $26|19 - y$, i.e. $26k = 19 - y$ (for $k \in \mathbb{Z}$) i.e. $y = 19 - 26 \cdot k$ for any $k \in \mathbb{Z}$

So $\{\dots, -7, 19, 45, \dots 19 + 26k, \dots\}$

Multiplicative Inverse

The number b is a multiplicative inverse of a (mod n) if $ba \equiv 1(mod\ n)$.

If gcd$(a, n) = 1$ then the multiplicative inverse exists.

If gcd$(a, n) \neq 1$ then the inverse does not exist.

Arithmetic $(mod\ p)$ for p prime is really nice for that reason.

Sometimes equivalences still have solutions when you don't have inverses (but sometimes they don't) – you'll experiment with these facts on HW5.

Proof By Contradiction

Proof By Contradiction

Suppose the negation of your claim.

Show that you can derive `False` (i.e. (¬claim) → `F`)

If your proof is right, the implication is true.

So ¬claim must be `False`.

So claim must be `True`!

Proof By Contradiction

Claim: $\sqrt{2}$ is irrational (i.e. not rational).

Proof:

Proof By Contradiction

Claim: $\sqrt{2}$ is irrational (i.e. not rational).

Proof:

Suppose for the sake of contradiction that $\sqrt{2}$ is rational.

But [] is a contradiction!

We don't have a fixed target.
That can make this proof harder.

Proof By Contradiction

If a^2 is even then a is even.

Claim: $\sqrt{2}$ is irrational (i.e. not rational).

Proof:

Suppose for the sake of contradiction that $\sqrt{2}$ is rational.

By definition of rational, there are integers s, t such that $t \neq 0$ and $\sqrt{2} = s/t$

Let $p = \frac{s}{\gcd(s,t)}, q = \frac{t}{\gcd(s,t)}$ Note that $\gcd(p, q) = 1$.

$\sqrt{2} = \frac{p}{q}$

That's is a contradiction! We conclude $\sqrt{2}$ is irrational.

Proof By Contradiction

If a^2 is even then a is even.

Claim: $\sqrt{2}$ is irrational (i.e. not rational).

Proof:

Suppose for the sake of contradiction that $\sqrt{2}$ is rational.

By definition of rational, there are integers s, t such that $\mathrm{t} \neq 0$ and $\sqrt{2} = s/t$

Let $p = \frac{\mathrm{s}}{\mathrm{gcd(s,t)}}, \mathrm{q} = \frac{\mathrm{t}}{\mathrm{gcd(s,t)}}$ Note that $\mathrm{gcd}(p, q) = 1$.

$\sqrt{2} = \frac{p}{q}$

$2 = \frac{p^2}{q^2}$

$2q^2 = p^2$ so p^2 is even.

That’s is a contradiction! We conclude $\sqrt{2}$ is irrational.

Proof By Contradiction

If a^2 is even then a is even.

Claim: $\sqrt{2}$ is irrational (i.e. not rational).

Proof:

Suppose for the sake of contradiction that $\sqrt{2}$ is rational.

By definition of rational, there are integers s, t such that $\mathrm{t} \neq 0$ and $\sqrt{2} = s/t$

Let $p = \frac{\mathrm{s}}{\mathrm{gcd(s,t)}}, \mathrm{q} = \frac{\mathrm{t}}{\mathrm{gcd(s,t)}}$ Note that $\mathrm{gcd}(p, q) = 1.$

$\sqrt{2} = \frac{p}{q}$

$2 = \frac{p^2}{q^2}$

$2q^2 = p^2$ so p^2 is even. By the fact above, p is even, i.e. $p = 2k$ for some integer k. Squaring both sides $p^2 = 4k^2$

Substituting into our original equation, we have: $2q^2 = 4k^2$, i.e. $q^2 = 2k^2$.

So q^2 is even. Applying the fact above again, q is even.

But if both p and q are even, $\mathrm{gcd}(p, q) \geq 2$. But we said $\mathrm{gcd}(p, q) = 1$

That's is a contradiction! We conclude $\sqrt{2}$ is irrational.

Proof By Contradiction

How in the world did we know how to do that?

In real life…lots of attempts that didn’t work.

Be very careful with proof by contradiction – without a clear target, you can easily end up in a loop of trying random things and getting nowhere.

What’s the difference?

What’s the difference between proof by contrapositive and proof by contradiction?

Show $p \to q$	**Proof by contradiction**	**Proof by contrapositive**
Starting Point	$\neg(p \to q) \equiv (p \wedge \neg q)$	$\neg q$
Target	Something false	$\neg p$

Show p	**Proof by contradiction**	**Proof by contrapositive**
Starting Point	$\neg p$	---
Target	Something false	---

Another Proof By Contradiction

Claim: There are infinitely many primes.

Proof:

Another Proof By Contradiction

Claim: There are infinitely many primes.

Proof:

Suppose for the sake of contradiction, that there are only finitely many primes. Call them $p_1, p_2, \dots, p_k$.

But [] is a contradiction! So there must be infinitely many primes.

Another Proof By Contradiction

Claim: There are infinitely many primes.

Proof:

Suppose for the sake of contradiction, that there are only finitely many primes. Call them $p_1, p_2, \dots, p_k$.

Consider the number $q = p_1 \cdot p_2 \cdot \dots \cdot p_k + 1$

Case 1: q is prime

Case 2: q is composite

But [] is a contradiction! So there must be infinitely many primes.

Another Proof By Contradiction

Claim: There are infinitely many primes.

Proof:

Suppose for the sake of contradiction, that there are only finitely many primes. Call them $p_1, p_2, \dots, p_k$.

Consider the number $q = p_1 \cdot p_2 \cdot \cdots \cdot p_k + 1$

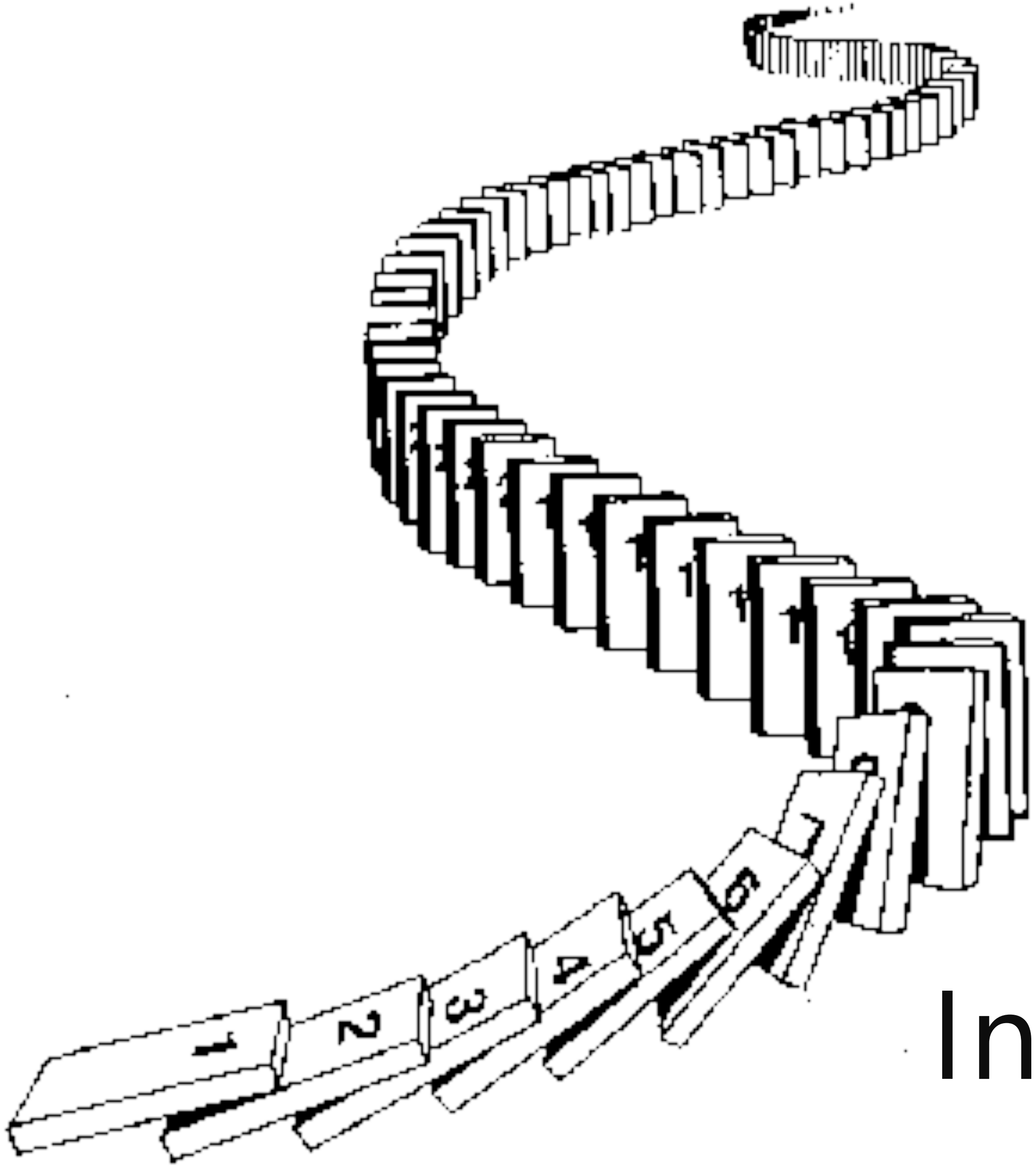
Case 1: q is prime

$q > p_i$ for all i. But every prime was supposed to be on the list $p_1, \dots, p_k$. A contradiction!

Case 2: q is composite

Some prime on the list (say p_i) divides q. So $q\%p_i = 0$. and $(p_1 p_2 \cdots p_k + 1)\%p_i = 1$. But $q = (p_1 p_2 \cdots p_k + 1)$. That's a contradiction!

In either case we have a contradiction! So there must be infinitely many primes.



Induction

CSE 311 Autumn 20
Lecture 14

Why does recursion work?

Fill out the poll everywhere for Activity Credit!
Go to pollev.com/cse311 and login with your UW identity
Or text cse311 to 22333

```
//Assume i is a nonnegative integer
//returns 2^i.
public int CalculatesTwoToTheI(int i){
     if(i == 0)
          return 1;
     else
          return 2*CaclulatesTwoToTheI(i-1);
}
```

Why does `CalculatesTwoToTheI(4)` calculate `2^4?`

`Convince the other people in your room`

Why does recursion work?

Something like this:

Well, as long as `CalculatesTwoToTheI(3) = 8`, we get 16…

Which happens as long as `CalculatesTwoToTheI(2) = 4`

Which happens as long as `CalculatesTwoToTheI(1) = 2`

Which happens as long as `CalculatesTwoToTheI(0) = 1`

And it is! Because that’s what the base case says.

Why does recursion work?

There's really only two cases.

The Base Case is Correct

`CalculatesTwoToTheI(0) = 1 (which it should!)`

And that means `CalculatesTwoToTheI(1) = 2`, (like it should)

And that means `CalculatesTwoToTheI(2) = 4`, (like it should)

And that means `CalculatesTwoToTheI(3) = 8,` (like it should)

And that means `CalculatesTwoToTheI(4) = 16,` (like it should)

IF the recursive call we make is correct
THEN our value is correct.

Why does recursion work?

The code has two big cases,

So our proof had two big cases

“The base case of the code produces the correct output”

“IF the calls we rely on produce the correct output THEN the current call produces the right output”

A bit more formally…

“The base case of the code produces the correct output”

“IF the calls we rely on produce the correct output THEN the current call produces the right output”

Let $P(i)$ be “`CalculatesTwoToTheI(i)`” returns 2^i.

How do we know $P(4)$?

$P(0)$ is true.

And $P(0) \rightarrow P(1)$, so $P(1)$.

And $P(1) \rightarrow P(2)$, so $P(2)$.

And $P(2) \rightarrow P(3)$, so $P(3)$.

And $P(3) \rightarrow P(4)$, so $P(4)$.

A bit more formally…

This works alright for $P(4)$.

What about $P(1000)$? $P(1000000000)$?

At this point, we'd need to show that implication $P(k) \rightarrow P(k+1)$ for A BUNCH of values of k.

But the code is the same each time.

And so was the argument!

We should instead show $\forall k[P(k) \rightarrow P(k+1)]$.

Induction

Your new favorite proof technique!

How do we show $\forall n, P(n)$?

Show $P(0)$

Show $\forall k(P(k) \rightarrow P(k+1))$

Induction

```
//Assume i is a nonnegative integer
public int CalculatesTwoToTheI(int i){
       if(i == 0)
              return 1;
       else
              return 2*CaclulatesTwoToTheI(i-1);
}
```

Let $P(i)$ be “`CalculatesTwoToTheI(i)`” returns 2^i.

Note that if the input i is 0, then the if-statement evaluates to true, and $1 = 2{}^\wedge 0$ is returned, so $P(0)$ is true.

Suppose $P(k)$ holds for an arbitrary $k \geq \mathbf{0}$.

So $P(k+1)$ holds.

Therefore $P(n)$ holds for all $n \geq \mathbf{0}$ by the principle of induction.

Making Induction Proofs Pretty

Let $P(i)$ be "`CalculatesTwoToTheI(i)`" returns 2^i.

Base Case $(i = 0)$ Note that if the input i is 0, then the if-statement evaluates to true, and $1 = 2^\wedge 0$ is returned, so $P(0)$ is true.

Inductive Hypothesis: Suppose $P(k)$ holds for an arbitrary $k \geq 0$.

Inductive Step: Since $k \geq 0, k \geq 1$, so the code goes to the recursive case. We will return $2 \cdot$ `CalculatesTwoToTheI(k)`. By Inductive Hypothesis,

`CalculatesTwoToTheI(k)` = 2^k. Thus we return $2 \cdot 2^k = 2^{k+1}$.

So $P(k+1)$ holds.

Therefore $P(n)$ holds for all $n \geq 0$ by the principle of induction.

Making Induction Proofs Pretty

All of our induction proofs will come in 5 easy(?) steps!

1. Define $P(n)$. State that your proof is by induction on n.
2. Show $P(0)$ i.e. show the base case
3. Suppose $P(k)$ for an arbitrary k.
4. Show $P(k+1)$ (i.e. get $P(k) \rightarrow P(k+1)$)
5. Conclude by saying $P(n)$ is true for all n by induction.

Some Other Notes

Always state where you use the inductive hypothesis when you're using it in the inductive step.

It's usually the key step, and the reader really needs to focus on it.

Be careful about what values you're assuming the Inductive Hypothesis for – the smallest possible value of k should assume the base case but nothing more.

The Principle of Induction (formally)

Principle of Induction

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

Informally: if you knock over one domino, and every domino knocks over the next one, then all your dominoes fell over.

More Induction

Induction doesn’t **only** work for code!

Show that $\sum_{i=0}^{n} 2^i = 1 + 2 + 4 + \cdots + 2^n = 2^{n+1} - 1$.

More Induction

Induction doesn't **only** work for code!

Show that $\sum_{i=0}^{n} 2^i = 1 + 2 + 4 + \cdots + 2^n = 2^{n+1} - 1$.

Let $P(n) =$"$\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$."

We show $P(n)$ holds for all n by induction on n.

Base Case (   )

Inductive Hypothesis:

Inductive Step:

$P(n)$ holds for all $n \geq 0$ by the principle of induction.

More Induction

Induction doesn't **only** work for code!

Show that $\sum_{i=0}^{n} 2^i = 1 + 2 + 4 + \cdots + 2^n = 2^{n+1} - 1$.

Let $P(n) =$"$\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$."

We show $P(n)$ holds for all n by induction on n.

Base Case $(n = 0)$ $\sum_{i=0}^{0} 2^i = 1 = 2 - 1 = 2^{0+1} - 1$.

Inductive Hypothesis: Suppose $P(k)$ holds for an arbitrary $k \geq 0$.

Inductive Step: We show $P(k+1)$. Consider the summation$\sum_{i=0}^{k+1} 2^i = 2^{\mathrm{k}+1} + \sum_{i=0}^{k} 2^i = 2^{k+1} + 2^{k+1} - 1$, where the last step is by IH.

Simplifying, we get: $\sum_{i=0}^{k+1} 2^i = 2^{k+1} + 2^{k+1} - 1 = 2 \cdot 2^{k+1} - 1 = 2^{(k+1)+1} - 1$.

$P(n)$ holds for all $n \geq 0$ by the principle of induction.