Mathematical Induction

Part One

Everybody – do the wave!

The Wave

- If done properly, everyone will eventually end up joining in.
- Why is that?
  - Someone (me!) started everyone off.
  - Once the person before you did the wave, you did the wave.

Let P be some predicate. The ***principle of mathematical induction*** states that if

If it starts true…

$$P(0) \text{ is true}$$

…and it stays true…

and

$$\forall k \in \mathbb{N}.\ (P(k) \to P(k+1))$$

then

$$\forall n \in \mathbb{N}.\ P(n)$$

…then it's always true.

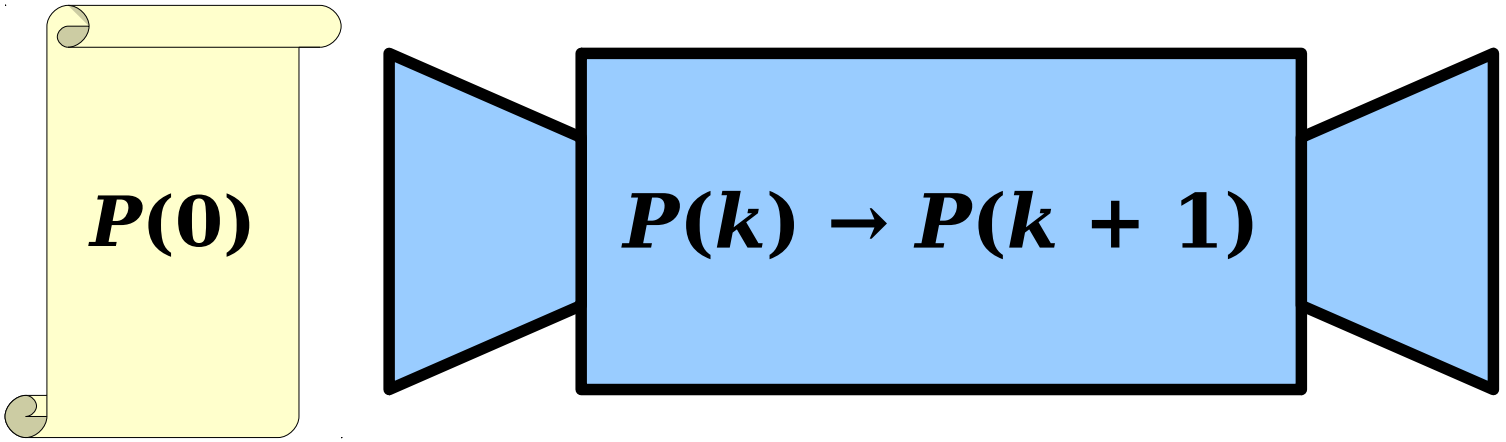
Induction, Intuitively

$$P(0)$$

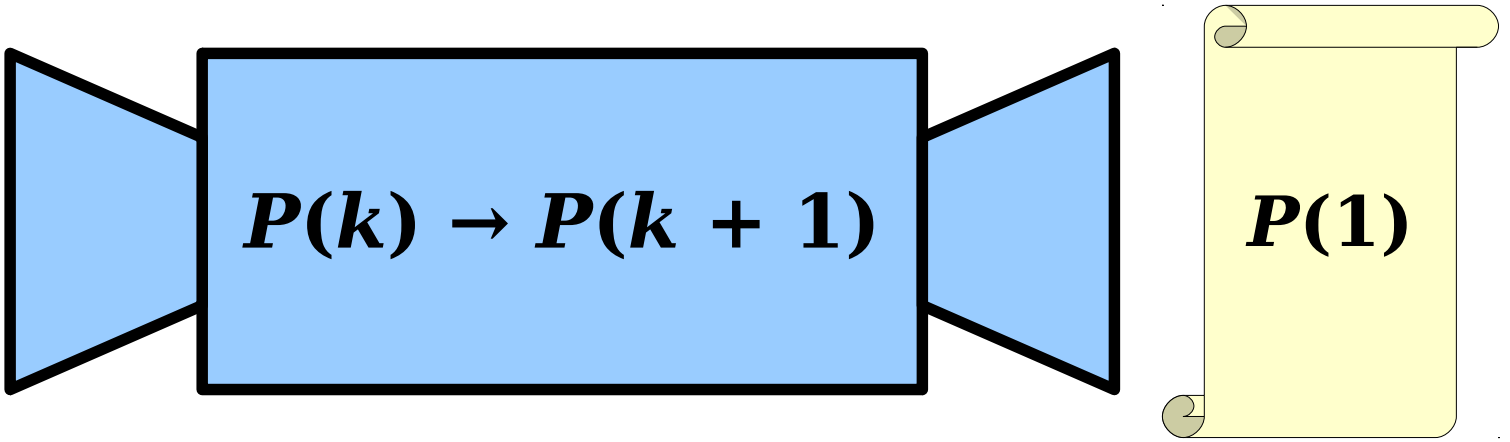
$$\forall k \in \mathbb{N}.\ (P(k) \to P(k+1))$$

- It's true for 0.
- Since it's true for 0, it's true for 1.
- Since it's true for 1, it's true for 2.
- Since it's true for 2, it's true for 3.
- Since it's true for 3, it's true for 4.
- Since it's true for 4, it's true for 5.
- Since it's true for 5, it's true for 6.
- …

Why Induction Works



Why Induction Works



Proof by Induction

- A ***proof by induction*** is a way to use the principle of mathematical induction to show that some result is true for all natural numbers n.
- In a proof by induction, there are three steps:
  - Prove that $P(0)$ is true.
    - This is called the ***basis*** or the ***base case***.
  - Prove that if $P(k)$ is true, then $P(k+1)$ is true.
    - This is called the ***inductive step***.
    - The assumption that $P(k)$ is true is called the ***inductive hypothesis***.
  - Conclude, by induction, that $P(n)$ is true for all $n \in \mathbb{N}$.

Some Sums

$$2^0 = 1 \qquad = 2^1 - 1$$

$$2^0 + 2^1 = 1 + 2 = 3 = 2^2 - 1$$

$$2^0 + 2^1 + 2^2 = 1 + 2 + 4 = 7 = 2^3 - 1$$

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

$$2^0 + 2^1 + 2^2 + 2^3 + 2^4 = 1 + 2 + 4 + 8 + 16 = 31 = 2^5 - 1$$

Theorem: The sum of the first n powers of two is $2^n - 1$.

Proof: Let $P(n)$ be the statement "the sum of the first n powers of two is $2^n - 1$." We will prove, by induction, that $P(n)$ is true for all $n \in \mathbb{N}$, from which the theorem follows.

At the start of the proof, we tell the reader what predicate we're going to show is true for all natural numbers n, then tell them we're going to prove it by induction.

Theorem: The sum of the first n powers of two is $2^n - 1$.

Proof: Let $P(n)$ be the statement "the sum of the first n powers of two is $2^n - 1$." We will prove, by induction, that $P(n)$ is true for all $n \in \mathbb{N}$, from which the theorem follows.

In a proof by induction, we need to prove that

- $P(0)$ is true
- If $P(k)$ is true, then $P(k+1)$ is true.

Theorem: The sum of the first n powers of two is $2^n - 1$.

Proof: Let $P(n)$ be the statement "the sum of the first n powers of two is $2^n - 1$." We will prove, by induction, that $P(n)$ is true for all $n \in \mathbb{N}$, from which the theorem follows.

For our base case, we need to show $P(0)$ is true, meaning that the sum of the first zero powers of two is $2^0 - 1$.

Here, we state what $P(0)$ actually says. Now, can go prove this using any proof techniques we'd like!

Theorem: The sum of the first n powers of two is $2^n - 1$.

Proof: Let $P(n)$ be the statement "the sum of the first n powers of two is $2^n - 1$." We will prove, by induction, that $P(n)$ is true for all $n \in \mathbb{N}$, from which the theorem follows.

For our base case, we need to show $P(0)$ is true, meaning that the sum of the first zero powers of two is $2^0 - 1$. Since the sum of the first zero powers of two is zero and $2^0 - 1$ is zero as well, we see that $P(0)$ is true.

In a proof by induction, we need to prove that

- ✓ $P(0)$ is true
- ☐ If $P(k)$ is true, then $P(k+1)$ is true.

Theorem: The sum of the first n powers of two is $2^n - 1$.

Proof: Let $P(n)$ be the statement "the sum of the first n powers of two is $2^n - 1$." We will prove, by induction, that $P(n)$ is true for all $n \in \mathbb{N}$, from which the theorem follows.

Fo ing
th Since
th - 1
is

In a proof by induction, we need to prove that

- ✓ $P(0)$ is true
- □ If $P(k)$ is true, then $P(k+1)$ is true.

What should the next step of this proof be?

A. Prove that, for any $k \in \mathbb{N}$, that $P(k)$ is true.
B. Assume for any $k \in \mathbb{N}$ that $P(k)$ and $P(k+1)$ are true.
C. Assume that $P(k)$ holds for all $k \in \mathbb{N}$.
D. Pick an arbitrary $k \in \mathbb{N}$, and prove $P(k+1)$.
E. Pick an arbitrary $k \in \mathbb{N}$, assume $P(k)$, and prove $P(k+1)$.
F. None of these, or more than one of these.

Answer at **PollEv.com/cs103** or
text **CS103** to **22333** once to join, then **A**, …, or **E**.

Theorem: The sum of the first n powers of two is $2^n - 1$.

Proof: Let $P(n)$ be the statement "the sum of the first n powers of two is $2^n - 1$." We will prove, by induction, that $P(n)$ is true for all $n \in \mathbb{N}$, from which the theorem follows.

For our base case, we need to show $P(0)$ is true, meaning that the sum of the first zero powers of two is $2^0 - 1$. Since the sum of the first zero powers of two is zero and $2^0 - 1$ is zero as well, we see that $P(0)$ is true.

For the inductive step, assume that for some arbitrary $k \in \mathbb{N}$ that $P(k)$ holds, meaning that

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

We need to show that $P(k + 1)$ holds, meaning that the sum of the first $k + 1$ powers of two is $2^{k+1} - 1$.

The goal of this step is to prove

"If $P(k)$ is true, then $P(k+1)$ is true."

To do this, we'll choose an arbitrary k, assume that $P(k)$ is true, then try to prove $P(k+1)$.

Theorem: The sum of the first n powers of two is $2^n - 1$.

Proof: Let $P(n)$ be the statement "the sum of the first n powers of two is $2^n - 1$." We will prove, by induction, that $P(n)$ is true for all $n \in \mathbb{N}$, from which the theorem follows.

For our base case, we need to show $P(0)$ is true, meaning that the sum of the first zero powers of two is $2^0 - 1$. Since the sum of the first zero powers of two is zero and $2^0 - 1$ is zero as well, we see that $P(0)$ is true.

For the inductive step, assume that for some arbitrary $k \in \mathbb{N}$ that $P(k)$ holds, meaning that

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

We need to show that $P(k + 1)$ holds, meaning that the sum of the first $k + 1$ powers of two is $2^{k+1} - 1$.

Here, we explicitly state $P(k+1)$, which is what we want to prove. Now, we can use any proof technique we want to prove it.

Theorem: The sum of the first n powers of two is $2^n - 1$.

Proof: Let $P(n)$ be the statement "the sum of the first n powers of two is $2^n - 1$." We will prove, by induction, that $P(n)$ is true f s.

For ou meaning that t 1. Since the su d $2^0 - 1$ is zer

Here, we use our inductive hypothesis (the assumption that $P(k)$ is true) to simplify a complex expression. This is a common theme in inductive proofs.

For the inductive step, assume that for some arbitrary $k \in \mathbb{N}$ that $P(k)$ holds, meaning that

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

We need to show that $P(k + 1)$ holds, meaning that the sum of the first $k + 1$ powers of two is $2^{k+1} - 1$. To see this, notice that

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

Theorem: The sum of the first n powers of two is $2^n - 1$.

Proof: Let $P(n)$ be the statement "the sum of the first n powers of two is $2^n - 1$." We will prove, by induction, that $P(n)$ is true for all $n \in \mathbb{N}$, from which the theorem follows.

For our base case, we need to show $P(0)$ is true, meaning that the sum of the first zero powers of two is $2^0 - 1$. Since the sum of the first zero powers of two is zero and $2^0 - 1$ is zero as well, we see that $P(0)$ is true.

For the inductive step, assume that for some arbitrary $k \in \mathbb{N}$ that $P(k)$ holds, meaning that

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

We need to show that $P(k + 1)$ holds, meaning that the sum of the first $k + 1$ powers of two is $2^{k+1} - 1$. To see this, notice that

$$\begin{aligned} 2^0 + 2^1 + \ldots + 2^{k-1} + 2^k &= (2^0 + 2^1 + \ldots + 2^{k-1}) + 2^k \\ &= 2^k - 1 + 2^k \qquad \textit{(via (1))} \\ &= 2(2^k) - 1 \\ &= 2^{k+1} - 1. \end{aligned}$$

Therefore, $P(k + 1)$ is true, completing the induction. ■

A Quick Aside

- This result helps explain the range of numbers that can be stored in an `int`.
- If you have an unsigned 32-bit integer, the largest value you can store is given by $1 + 2 + 4 + 8 + \ldots + 2^{31} = 2^{32} - 1$.
- This formula for sums of powers of two has many other uses as well. If we have time, we'll see one more today.
  - If not, we'll see it next time!

Structuring a Proof by Induction

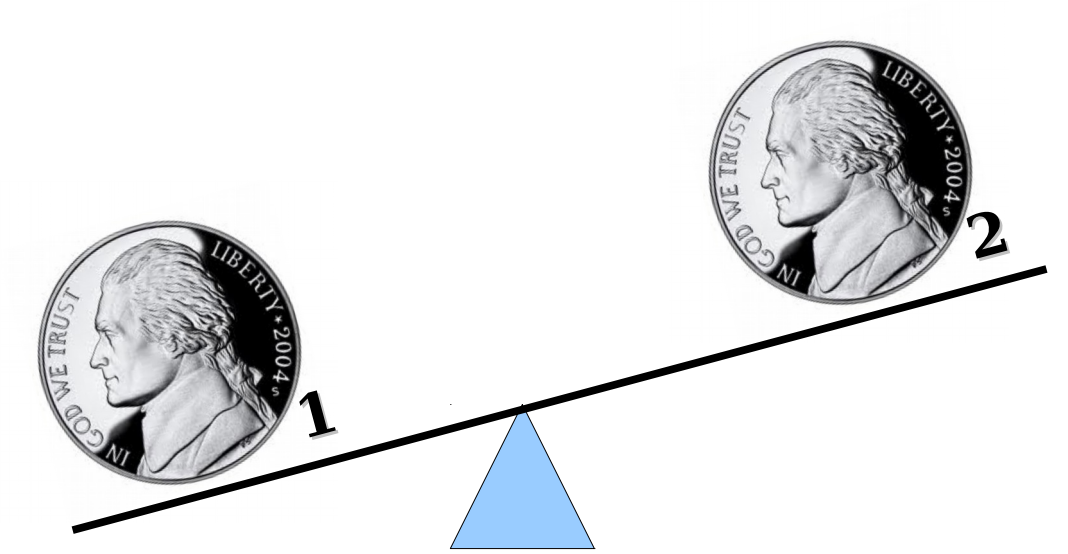
- Define some predicate P that you'll show, by induction, is true for all natural numbers.
- Prove the base case:
  - State that you're going to prove that $P(0)$ is true, then go prove it.
- Prove the inductive step:
  - Say that you're assuming $P(k)$ for some arbitrary natural number k, then write out exactly what that means.
  - Say that you're going to prove $P(k+1)$, then write out exactly what that means.
  - Prove that $P(k+1)$ using any proof technique you'd like!
- This is a rather verbose way of writing inductive proofs. As we get more experience with induction, we'll start leaving out some details from our proofs.

The Counterfeit Coin Problem

Problem Statement

- You are given a set of three seemingly identical coins, two of which are real and one of which is counterfeit.
- The counterfeit coin weighs more than the rest of the coins.
- You are given a balance. Using only one weighing on the balance, find the counterfeit coin.

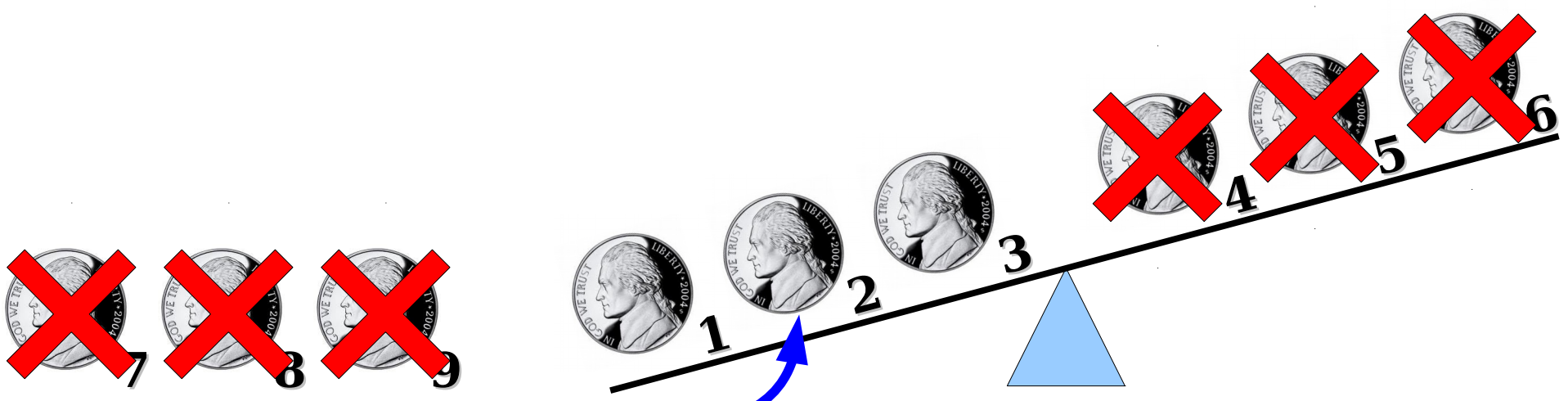
Finding the Counterfeit Coin



A Harder Problem

- You are given a set of ***nine*** seemingly identical coins, eight of which are real and one of which is counterfeit.
- The counterfeit coin weighs more than the rest of the coins.
- You are given a balance. Using only ***two*** weighings on the balance, find the counterfeit coin.

Finding the Counterfeit Coin



Now we have one weighing to find the counterfeit out of these three coins.

Can we generalize this?

A Pattern

- Assume out of the coins that are given, exactly one is counterfeit and weighs more than the other coins.
- If we have no weighings, how many coins can we have while still being able to find the counterfeit?
  - ***One*** coin, since that coin has to be the counterfeit!
- If we have one weighing, we can find the counterfeit out of ***three*** coins.
- If we have two weighings, we can find the counterfeit out of ***nine*** coins.

So far, we have

$1, 3, 9 = 3^0, 3^1, 3^2$

Does this pattern continue?

Theorem: If exactly one coin in a group of 3^n coins is heavier than the rest, that coin can be found using only n weighings on a balance.

Proof: Let $P(n)$ be ...

Which of these is a good choice for $P(n)$?

A. 3^n.

B. A group of 3^n coins.

C. For any $n \in \mathbb{N}$, if there are 3^n coins of which one is heavier than the rest, we can find it using n weighings on a balance.

D. If we can find the heavier coin out of a group of 3^n coins in n weighings, then we can find the heavier coin out of a group of 3^{n+1} coins in $n+1$ weighings.

E. None of these, or more than one of these.

Answer at **PollEv.com/cs103** or
text **CS103** to **22333** once to join, then **A**, ..., or **E**.

Theorem: If exactly one coin in a group of 3^n coins is heavier than the rest, that coin can be found using only n weighings on a balance.

Proof: Let $P(n)$ be the following statement:

> If exactly one coin in a group of 3^n coins is heavier than the rest, that coin can be found using only n weighings on a balance.

We'll use induction to prove that $P(n)$ holds for every $n \in \mathbb{N}$, from which the theorem follows.

At the start of the proof, we tell the reader what predicate we're going to show is true for all natural numbers n, then tell them we're going to prove it by induction.

Theorem: If exactly one coin in a group of 3^n coins is heavier than the rest, that coin can be found using only n weighings on a balance.

Proof: Let $P(n)$ be the following statement:

> If exactly one coin in a group of 3^n coins is heavier than the rest, that coin can be found using only n weighings on a balance.

We'll use induction to prove that $P(n)$ holds for every $n \in \mathbb{N}$, from which the theorem follows.

In a proof by induction, we need to prove that

- $P(0)$ is true
- If $P(k)$ is true, then $P(k+1)$ is true.

Theorem: If exactly one coin in a group of 3^n coins is heavier than the rest, that coin can be found using only n weighings on a balance.

Proof: Let $P(n)$ be the following statement:

If exactly one coin in a group of 3^n coins is heavier than the rest, that coin can be found using only n weighings on a balance.

We'll use induction to prove that $P(n)$ holds for every $n \in \mathbb{N}$, from which the theorem follows.

As our base case, we'll prove that $P(0)$ is true, meaning that if we have a set of $3^0=1$ coins with one coin heavier than the rest, we can find that coin with zero weighings.

Here, we state what $P(0)$ actually says. Now, can go prove this using any proof techniques we'd like!

Theorem: If exactly one coin in a group of 3^n coins is heavier than the rest, that coin can be found using only n weighings on a balance.

Proof: Let $P(n)$ be the following statement:

> If exactly one coin in a group of 3^n coins is heavier than the rest, that coin can be found using only n weighings on a balance.

We'll use induction to prove that $P(n)$ holds for every $n \in \mathbb{N}$, from which the theorem follows.

As our base case, we'll prove that $P(0)$ is true, meaning that if we have a set of $3^0=1$ coins with one coin heavier than the rest, we can find that coin with zero weighings. This is true because if we have just one coin, it's vacuously heavier than all the others, and no weighings are needed.

In a proof by induction, we need to prove that

- ✓ $P(0)$ is true
- ☐ If $P(k)$ is true, then $P(k+1)$ is true.

Theorem: If exactly one coin in a group of 3^n coins is heavier than the rest, that coin can be found using only n weighings on a balance.

Proof: Let $P(n)$ be the following statement:

If exactly one coin in a group of 3^n coins is heavier than the rest, that coin can be found using only n weighings on a balance.

We'll use induction to prove that $P(n)$ holds for every $n \in \mathbb{N}$, from which the theorem follows.

As our base case, we'll prove that $P(0)$ is true, meaning that if we have a set of $3^0=1$ coins with one coin heavier than the rest, we can find that coin with zero weighings. This is true because if we have just one coin, it's vacuously heavier than all the others, and no weighings are needed.

For the inductive step, suppose $P(k)$ is true for some arbitrary $k \in \mathbb{N}$, so we can find the heavier of 3^k coins in k weighings. We'll prove $P(k+1)$: that we can find the heavier of 3^{k+1} coins in $k+1$ weighings.

The goal of this step is to prove

"If $P(k)$ is true, then $P(k+1)$ is true."

To do this, we'll choose an arbitrary k, assume that $P(k)$ is true, then try to prove $P(k+1)$.

Theorem: If exactly one coin in a group of 3^n coins is heavier than the rest, that coin can be found using only n weighings on a balance.

Proof: Let $P(n)$ be the following statement:

If exactly one coin in a group of 3^n coins is heavier than the rest, that coin can be found using only n weighings on a balance.

We'll use induction to prove that $P(n)$ holds for every $n \in \mathbb{N}$, from which the theorem follows.

As our base case, we'll prove that $P(0)$ is true, meaning that if we have a set of $3^0=1$ coins with one coin heavier than the rest, we can find that coin with zero weighings. This is true because if we have just one coin, it's vacuously heavier than all the others, and no weighings are needed.

For the inductive step, suppose $P(k)$ is true for some arbitrary $k \in \mathbb{N}$, so we can find the heavier of 3^k coins in k weighings. We'll prove $P(k+1)$: that we can find the heavier of 3^{k+1} coins in $k+1$ weighings.

Here, we explicitly state $P(k+1)$, which is what we want to prove. Now, we can use any proof technique we want to try to prove it.

Theorem: If exactly one coin in a group of 3^n coins is heavier than the rest, that coin can be found using only n weighings on a balance.

Proof: Let $P(n)$ be the following statement:

If exact… rest, that c… ce.

We'll use … rom which the theore…

As our ba… if we have a set of 3… n find that coin with zero weighings. This is true because if we have just one coin, it's vacuously heavier than all the others, and no weighings are needed.

Here, we use our inductive hypothesis (the assumption that $P(k)$ is true) to solve this simpler version of the overall problem.

For the inductive step, suppose $P(k)$ is true for some arbitrary $k \in \mathbb{N}$, so we can find the heavier of 3^k coins in k weighings. We'll prove $P(k+1)$: that we can find the heavier of 3^{k+1} coins in $k+1$ weighings.

Suppose we have 3^{k+1} coins with one heavier than the others. Split the coins into three groups of 3^k coins each. Weigh two of the groups against one another. If one group is heavier than the other, the coins in that group must contain the heavier coin. Otherwise, the heavier coin must be in the group we didn't put on the scale. Therefore, with one weighing, we can find a group of 3^k coins containing the heavy coin. We can then use k more weighings to find the heavy coin in that group.

Theorem: If exactly one coin in a group of 3^n coins is heavier than the rest, that coin can be found using only n weighings on a balance.

Proof: Let $P(n)$ be the following statement:

> If exactly one coin in a group of 3^n coins is heavier than the rest, that coin can be found using only n weighings on a balance.

We'll use induction to prove that $P(n)$ holds for every $n \in \mathbb{N}$, from which the theorem follows.

As our base case, we'll prove that $P(0)$ is true, meaning that if we have a set of $3^0=1$ coins with one coin heavier than the rest, we can find that coin with zero weighings. This is true because if we have just one coin, it's vacuously heavier than all the others, and no weighings are needed.

For the inductive step, suppose $P(k)$ is true for some arbitrary $k \in \mathbb{N}$, so we can find the heavier of 3^k coins in k weighings. We'll prove $P(k+1)$: that we can find the heavier of 3^{k+1} coins in $k+1$ weighings.

Suppose we have 3^{k+1} coins with one heavier than the others. Split the coins into three groups of 3^k coins each. Weigh two of the groups against one another. If one group is heavier than the other, the coins in that group must contain the heavier coin. Otherwise, the heavier coin must be in the group we didn't put on the scale. Therefore, with one weighing, we can find a group of 3^k coins containing the heavy coin. We can then use k more weighings to find the heavy coin in that group.

We've given a way to use $k+1$ weighings and find the heavy coin out of a group of 3^{k+1} coins. Thus $P(k+1)$ is true, completing the induction. ■

Some Fun Problems

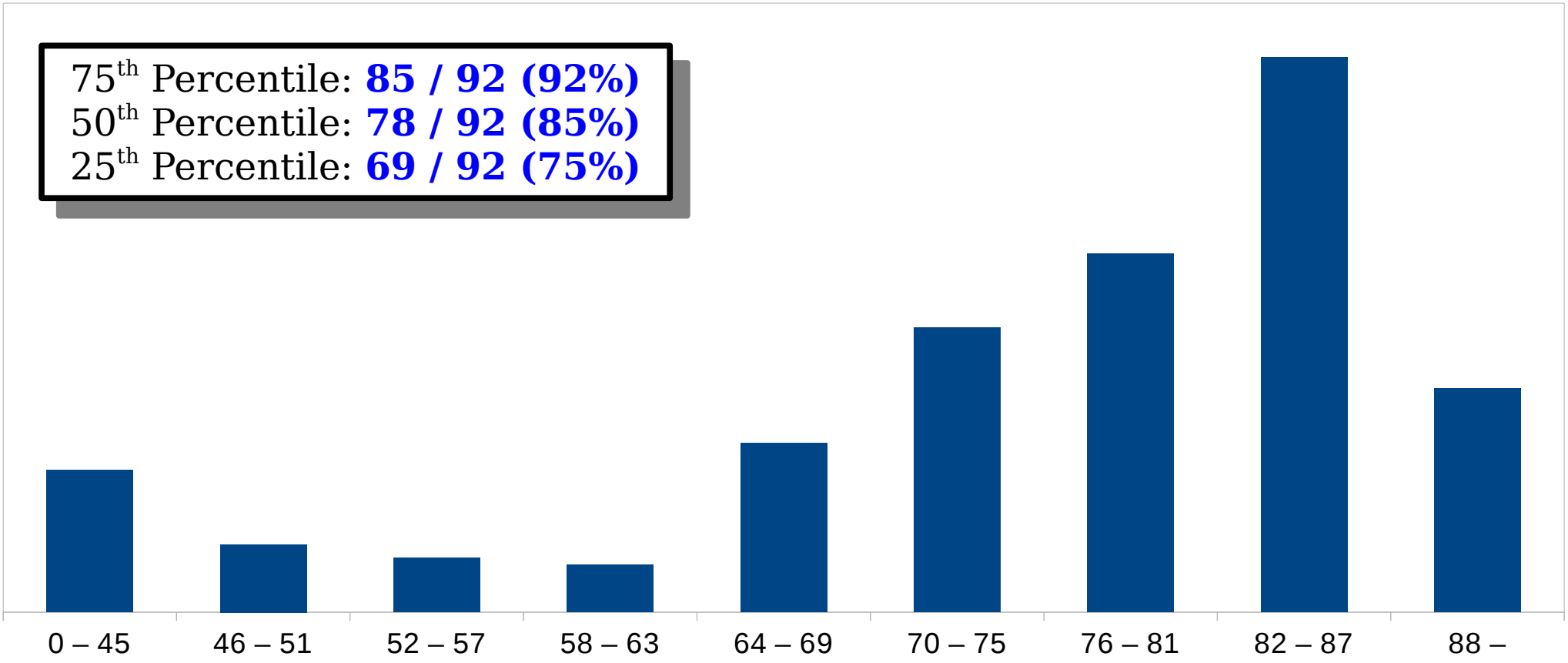
- Here's some nifty variants of this problem that you can work through:
  - Suppose that you have a group of coins where there's either exactly one heavier coin, or all coins weigh the same amount. If you only get k weighings, what's the largest number of coins where you can find the counterfeit or determine none exists?
  - What happens if the counterfeit can be either heavier or lighter than the other coins? What's the maximum number of coins where you can find the counterfeit if you have k weighings?
  - Can you find the counterfeit out of a group of more than 3^k coins with k weighings?
  - Can you find the counterfeit out of any group of at most 3^k coins with k weighings?

Time-Out for Announcements!

First Midterm Exam

- You’re done with the first midterm! Woohoo!
- We’ll be grading it over the weekend. Solutions will go out when we have context for them (common errors, stats, etc.), but feel free to ask us if you’re curious about anything!

Problem Set Three Grades



Problem Set Four

- Problem Set Four is due this Friday at 2:30PM.
- ***Recommendation:*** As soon as you can, review all the feedback you got on PS3 and from the PS4 checkpoint. Ask yourself these questions:
  - Based on the proofwriting and style feedback you received, do you know what specific changes you'd make to your answers?
  - If you made any logic errors, do you understand what those errors are to the point that you could explain them to someone else?
- Feel free to stop by office hours or to visit Piazza if you have questions. We're happy to help out! You can do this!

Back to CS103!

How Not To Induct

What's wrong with this proof?

Answer at **PollEv.com/cs103** or
text **CS103** to **22333** once to join, then your answer.

Theorem: The sum of the first n powers of two is 2^n.

Proof: Let $P(n)$ be the statement "the sum of the first n powers of two is 2^n." We will prove, by induction, that $P(n)$ is true for all $n \in \mathbb{N}$, from which the theorem follows.

For the inductive step, assume that for some arbitrary $k \in \mathbb{N}$ that $P(k)$ holds, meaning that

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

We need to show that $P(k + 1)$ holds, meaning that the sum of the first $k + 1$ powers of two is 2^{k+1}. To see this, notice that

$$\begin{aligned} 2^0 + 2^1 + \ldots + 2^{k-1} + 2^k &= (2^0 + 2^1 + \ldots + 2^{k-1}) + 2^k \\ &= 2^k + 2^k && \textit{(via (1))} \\ &= 2(2^k) \\ &= 2^{k+1}. \end{aligned}$$

Therefore, $P(k + 1)$ is true, completing the induction. ■

When writing a proof by induction,

make sure to prove the base case!

Otherwise, your argument is invalid!

Why did this work?

Theorem: The sum of the first n powers of two is 2^n.

Proof: Let $P(n)$ be the statement "the sum of the first n powers of two is 2^n." We will prove, by induction, that $P(n)$ is true for all $n \in \mathbb{N}$, from which the theorem follows.

For the inductive step, assume that for some arbitrary $k \in \mathbb{N}$ that $P(k)$ holds, meaning that

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

We need to show that $P(k + 1)$ holds, meaning that the sum of the first $k + 1$ powers of two is 2^{k+1}. To see this, notice that

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

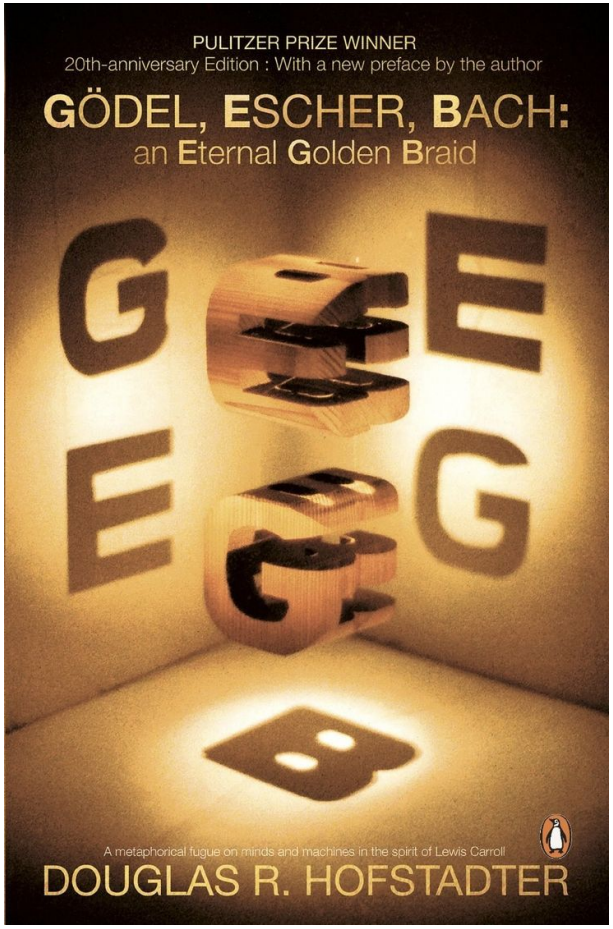
Therefore, $P(k + 1)$ is true, completing the induction. ∎

You can prove *anything* from a faulty assumption. This is called the **principle of explosion**. To see why, read "Animal, Vegetable, or Minister" for a silly example.

The MU Puzzle

Gödel, Escher Bach: An Eternal Golden Braid

- Douglas Hofstadter, cognitive scientist at the University of Indiana, wrote this Pulitzer-Prize-winning mind trip of a book.
- It's a great read after you've finished CS103 – you'll see so many of the ideas we'll cover presented in a totally different way!



The MU Puzzle

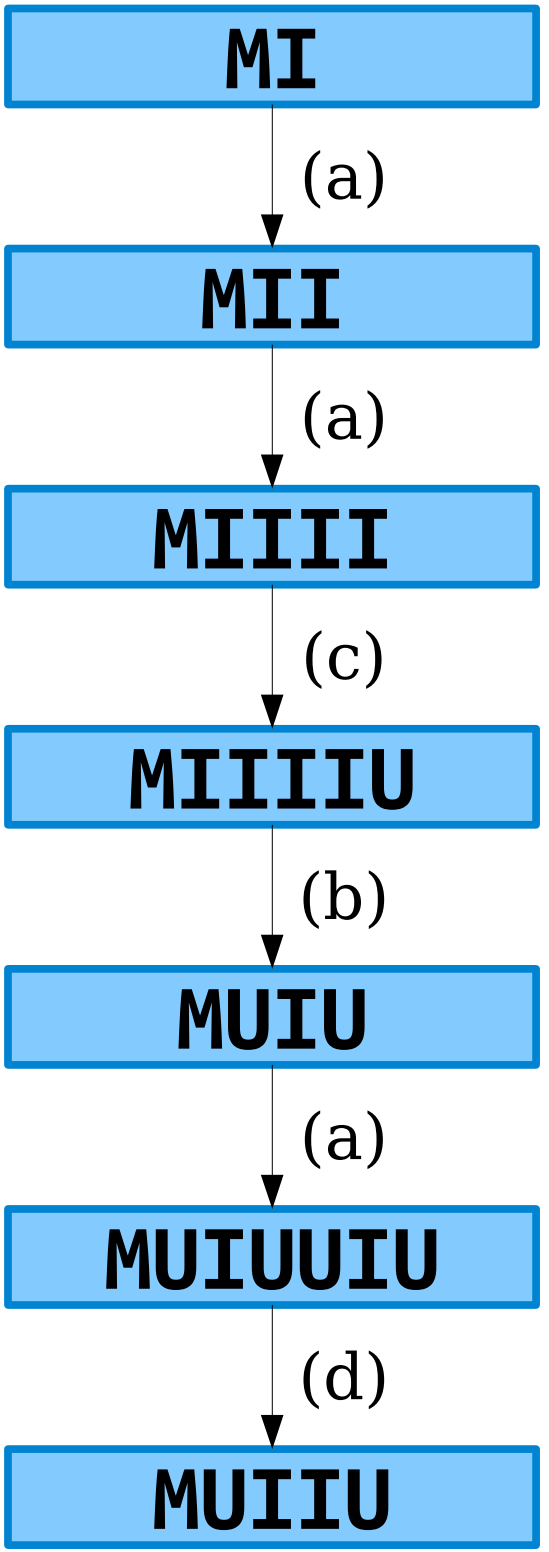
- Begin with the string `MI`.
- Repeatedly apply one of the following operations:
  - Double the contents of the string after the `M`: for example, `MIIU` becomes `MIIUIIU`, or `MI` becomes `MII`.
  - Replace `III` with `U`: `MIIII` becomes `MUI` or `MIU`.
  - Append `U` to the string if it ends in `I`: `MI` becomes `MIU`.
  - Remove any `UU`: `MUUU` becomes `MU`.
- ***Question***: How do you transform `MI` to `MU`?

(a) Double the string after an `M`.

(b) Replace `III` with `U`.

(c) Append `U`, if the string ends in `I`.

(d) Delete `UU` from the string.



Try It!

Starting with **MI**, apply these operations to make **MU**:

(a) Double the string after an **M**.

(b) Replace **III** with **U**.

(c) Append **U**, if the string ends in **I**.

(d) Delete **UU** from the string.

Not a single person in this room
was able to solve this puzzle.

Are we even sure that there is a solution?

Counting **I**'s


MI
1
MII
2
MIIII
4
MIIIIU
4
MIIIIUIIIIU
8
MIIIIUUIU
5
MIIIIUUIUIIIIUUIU
10
MUIUUIUIIIIUUIU
7

The Key Insight

- Initially, the number of `I`'s is *not* a multiple of three.
- To make `MU`, the number of `I`'s must end up as a multiple of three.
- Can we *ever* make the number of `I`'s a multiple of three?

Lemma 1: If n is an integer that is not a multiple of three, then $n - 3$ is not a multiple of three.

Proof: By contrapositive; we'll prove that if $n - 3$ is a multiple of three, then n is also a multiple of three. Because $n - 3$ is a multiple of three, we can write $n - 3 = 3k$ for some integer k. Then $n = 3(k+1)$, so n is also a multiple of three, as required. ■

Lemma 2: If n is an integer that is not a multiple of three, then $2n$ is not a multiple of three.

Proof: Let n be a number that isn't a multiple of three. If n is congruent to one modulo three, then $n = 3k + 1$ for some integer k. This means $2n = 2(3k+1) = 6k + 2 = 3(3k) + 2$, so $2n$ is not a multiple of three. Otherwise, n must be congruent to two modulo three, so $n = 3k + 2$ for some integer k. Then $2n = 2(3k+2) = 6k+4 = 3(2k+1) + 1$, and so $2n$ is not a multiple of three. ■

Lemma: No matter which moves are made, the number of **I**'s in the string never becomes multiple of three.

Proof: Let $P(n)$ be the statement "after any n moves, the number of **I**'s in the string will not be multiple of three." We will prove, by induction, that $P(n)$ is true for all $n \in \mathbb{N}$, from which the theorem follows.

As a base case, we'll prove $P(0)$, that the number of **I**'s after 0 moves is not a multiple of three. After no moves, the string is **MI**, which has one **I** in it. Since one isn't a multiple of three, $P(0)$ is true.

For our inductive step, suppose that $P(k)$ is true for some arbitrary $k \in \mathbb{N}$. We'll prove $P(k+1)$ is also true. Consider any sequence of $k+1$ moves. Let r be the number of **I**'s in the string after the kth move. By our inductive hypothesis (that is, $P(k)$), we know that r is not a multiple of three. Now, consider the four possible choices for the $k+1^{st}$ move:

Case 1: Double the string after the **M**. After this, we will have $2r$ **I**'s in the string, and from our lemma $2r$ isn't a multiple of three.

Case 2: Replace **III** with **U**. After this, we will have $r - 3$ **I**'s in the string, and by our lemma $r - 3$ is not a multiple of three.

Case 3: Either append **U** or delete **UU**. This preserves the number of **I**'s in the string, so we don't have a multiple of three **I**'s at this point.

Therefore, no sequence of $k+1$ moves ends with a multiple of three **I**'s. Thus $P(k+1)$ is true, completing the induction. ■

Theorem: The `MU` puzzle has no solution.

Proof: Assume for the sake of contradiction that the `MU` puzzle has a solution and that we can convert `MI` to `MU`. This would mean that at the very end, the number of `I`'s in the string must be zero, which is a multiple of three. However, we've just proven that the number of `I`'s in the string can never be a multiple of three.

We have reached a contradiction, so our assumption must have been wrong. Thus the `MU` puzzle has no solution. ■

Algorithms and Loop Invariants

- The proof we just made had the form
  - "If *P* is true before we perform an action, it is true after we perform an action."
- We could therefore conclude that after any series of actions of any length, if *P* was true beforehand, it is true now.
- In algorithmic analysis, this is called a ***loop invariant***.
- Proofs on algorithms often use loop invariants to reason about the behavior of algorithms.
  - Take CS161 for more details!

Next Time

- ***Variations on Induction***
  - Starting induction later.
  - Taking larger steps.
  - Complete induction.

Mathematical Induction

Part Two

Recap from Last Time

Let P be some predicate. The ***principle of mathematical induction*** states that if

If it starts true…

$$P(0) \text{ is true}$$

…and it stays true…

and

$$\forall k \in \mathbb{N}.\ (P(k) \to P(k+1))$$

then

$$\forall n \in \mathbb{N}.\ P(n)$$

…then it's always true.

New Stuff!

Theorem: The sum of the first n powers of two is $2^n - 1$.

Proof: Let $P(n)$ be the statement "the sum of the first n powers of two is $2^n - 1$." We will prove, by induction, that $P(n)$ is true for all $n \in \mathbb{N}$, from which the theorem follows.

For our base case, we need to show $P(0)$ is true, meaning that the sum of the first zero powers of two is $2^0 - 1$. Since the sum of the first zero powers of two is zero and $2^0 - 1$ is zero as well, we see that $P(0)$ is true.

For the inductive step, assume that for some arbitrary $k \in \mathbb{N}$ that $P(k)$ holds, meaning that

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

We need to show that $P(k + 1)$ holds, meaning that the sum of the first $k + 1$ powers of two is $2^{k+1} - 1$. To see this, notice that

$$\begin{aligned} 2^0 + 2^1 + \ldots + 2^{k-1} + 2^k &= (2^0 + 2^1 + \ldots + 2^{k-1}) + 2^k \\ &= 2^k - 1 + 2^k \qquad \textit{(via (1))} \\ &= 2(2^k) - 1 \\ &= 2^{k+1} - 1. \end{aligned}$$

Therefore, $P(k + 1)$ is true, completing the induction. ■

Induction in Practice

- Typically, a proof by induction will not explicitly state $P(n)$.
- Rather, the proof will describe $P(n)$ implicitly and leave it to the reader to fill in the details.
- Provided that there is sufficient detail to determine
  - what $P(n)$ is;
  - that $P(0)$ is true; and that
  - whenever $P(k)$ is true, $P(k+1)$ is true,

  the proof is usually valid.

Theorem: The sum of the first n powers of two is $2^n - 1$.

***Proof*:** By induction.

For our base case, we'll prove the theorem is true when $n = 0$. The sum of the first zero powers of two is zero, and $2^0 - 1 = 0$, so the theorem is true in this case.

For the inductive step, assume the theorem holds when $n = k$ for some arbitrary $k \in \mathbb{N}$. Then

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

So the theorem is true when $n = k+1$, completing the induction. ■

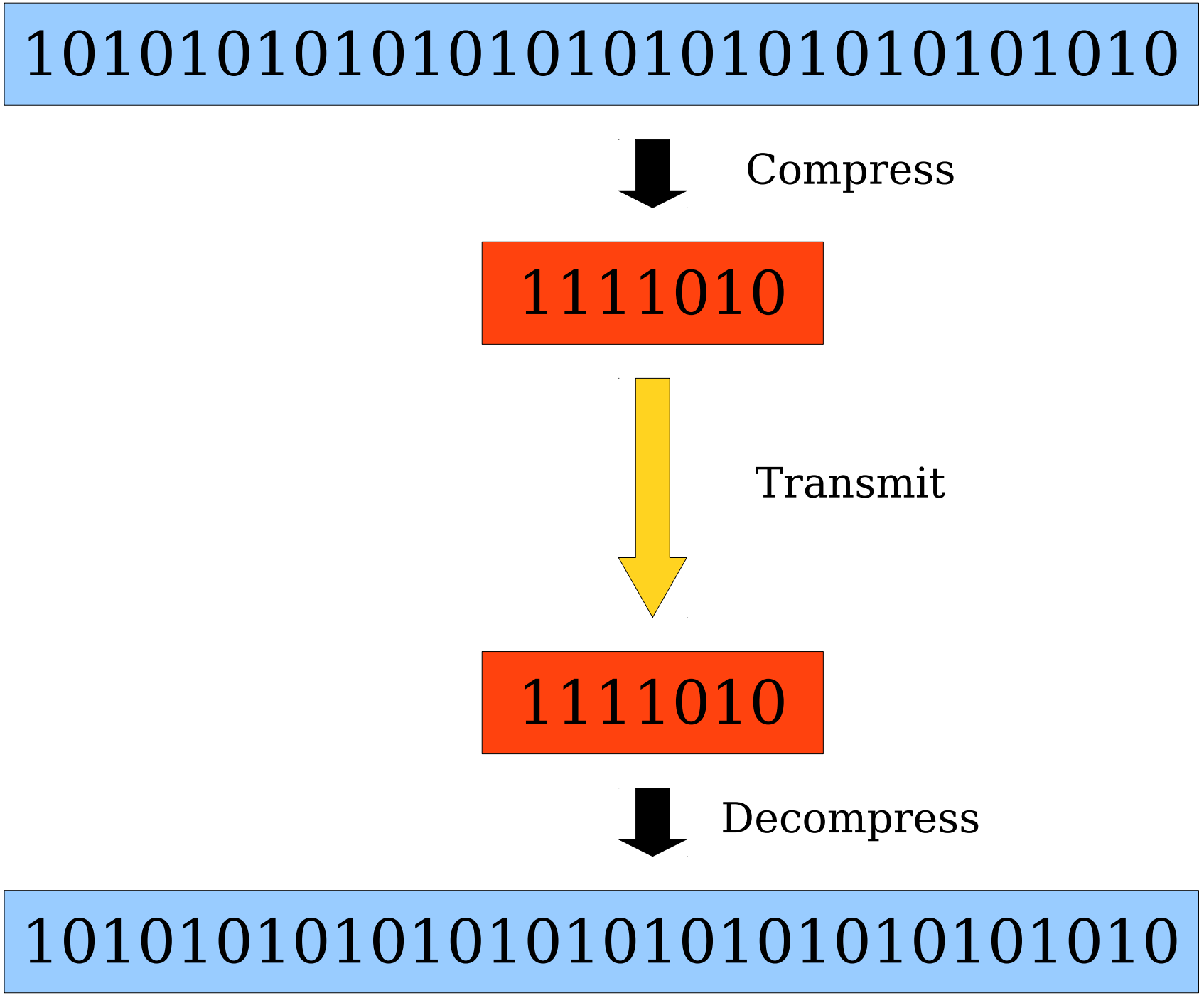
A Fun Application:
The Limits of Data Compression

Bitstrings

- A ***bitstring*** is a finite sequence of 0s and 1s.
- Examples:
  - 11011100
  - 010101010101
  - 0000
  - ε (the ***empty string***)
- There are 2^n bitstrings of length n.

Data Compression

- Inside a computer, all data are represented as sequences of 0s and 1s (bitstrings)
- To transfer data over a network (or on a flash drive, if you're still into that), it is useful to reduce the number of 0s and 1s before transferring it.
- Most real-world data can be compressed by exploiting redundancies.
  - Text repeats common patterns (“the”, “and”, etc.)
  - Bitmap images use similar colors throughout the image.
- ***Idea:*** Replace each bitstring with a *shorter* bitstring that contains all the original information.
  - This is called ***lossless data compression***.


10101010101010101010101010101010
Compress
1111010
Transmit
1111010
Decompress
10101010101010101010101010101010

Lossless Data Compression

- In order to losslessly compress data, we need two functions:
  - A ***compression function*** C, and
  - A ***decompression function*** D.
- We need to have $D(C(x)) = x$.
  - Otherwise, we can't uniquely encode or decode some bitstring.

How many of the following must be true about C and D?

C must be injective.
C must be surjective.
D must be injective.
D must be surjective.

Answer at **PollEv.com/cs103** or
text **CS103** to **22333** once to join, then a number.

Lossless Data Compression

- In order to losslessly compress data, we need two functions:
  - A ***compression function*** C, and
  - A ***decompression function*** D.
- We need to have $D(C(x)) = x$.
  - Otherwise, we can't uniquely encode or decode some bitstring.
- This means that D must be a left inverse of C, so (as you proved in PS3!) C must be injective.

A Perfect Compression Function

- Ideally, the compressed version of a bitstring would always be shorter than the original bitstring.
- ***Question***: Can we find a lossless compression algorithm that always compresses a string into a shorter string?
- To handle the issue of the empty string (which can't get any shorter), let's assume we only care about strings of length at least 10.

A Counting Argument

- Let $\mathbb{B}^n$ be the set of bitstrings of length n, and $\mathbb{B}^{<n}$ be the set of bitstrings of length less than n.
- How many bitstrings of length n are there?
  - ***Answer***: 2^n
- How many bitstrings of length *less than* n are there?
  - ***Answer***: $2^0 + 2^1 + \ldots + 2^{n-1} = 2^n - 1$
- By the pigeonhole principle, no function from $\mathbb{B}^n$ to $\mathbb{B}^{<n}$ can be injective – at least two elements must collide!
- Since a perfect compression function would have to be an injection from $\mathbb{B}^n$ to $\mathbb{B}^{<n}$, ***there is no perfect compression function!***

Why this Result is Interesting

- Our result says that no matter how hard we try, it is ***impossible*** to compress every string into a shorter string.
- No matter how clever you are, you cannot write a lossless compression algorithm that always makes strings shorter.
- In practice, only highly redundant data can be compressed.
- The fields of ***information theory*** and ***Kolmogorov complexity*** explore the limits of compression; if you're interested, go explore!

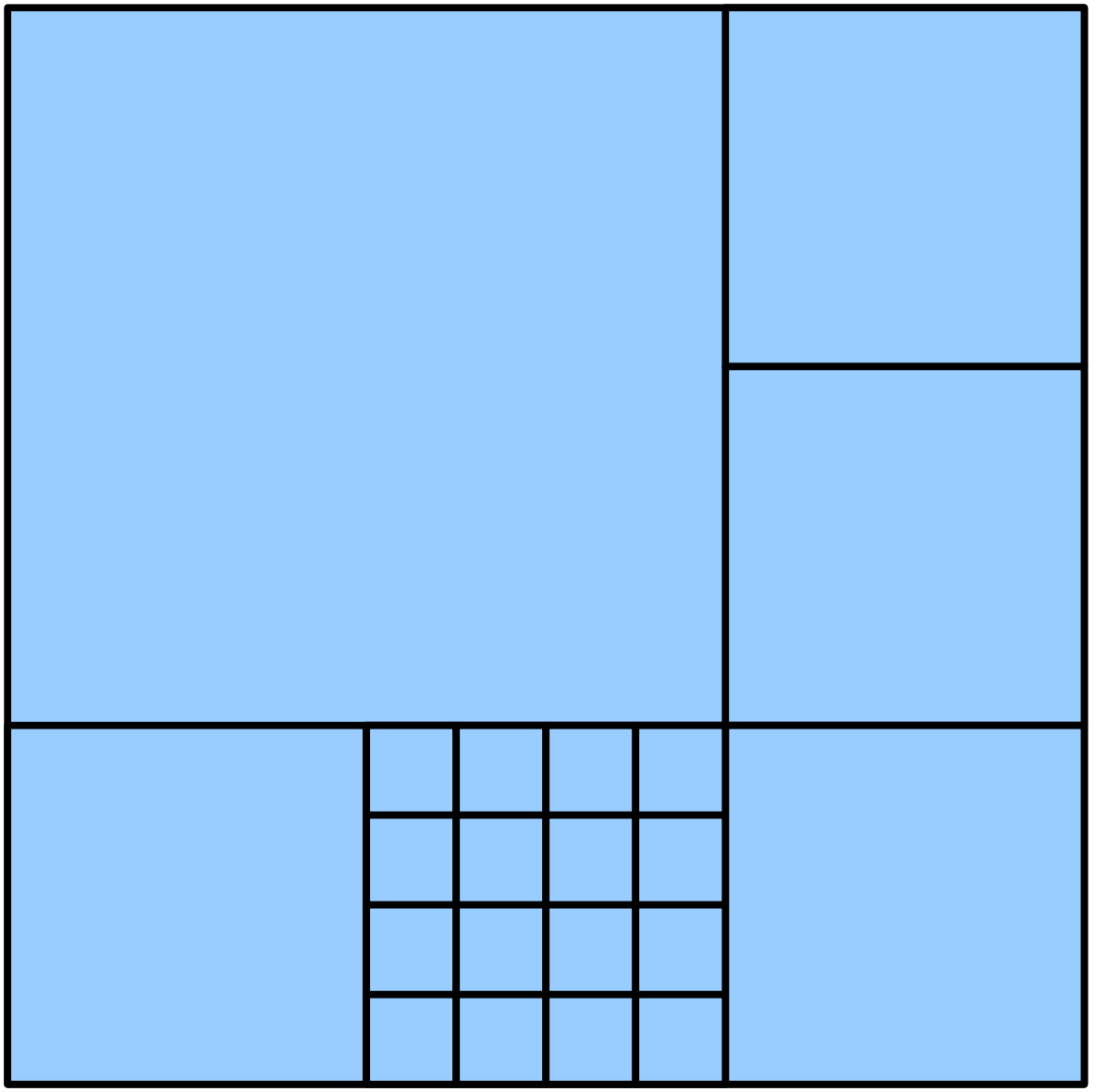
Variations on Induction: ***Starting Later***

Induction Starting at ***m***

- To prove that $P(n)$ is true for all natural numbers greater than or equal to ***m***:
  - Show that $P($***m***$)$ is true.
  - Show that for any $k \geq$ ***m***, that if $P(k)$ is true, then $P(k+1)$ is true.
  - Conclude $P(n)$ holds for all natural numbers greater than or equal to ***m***.

Variations on Induction: ***Bigger Steps***

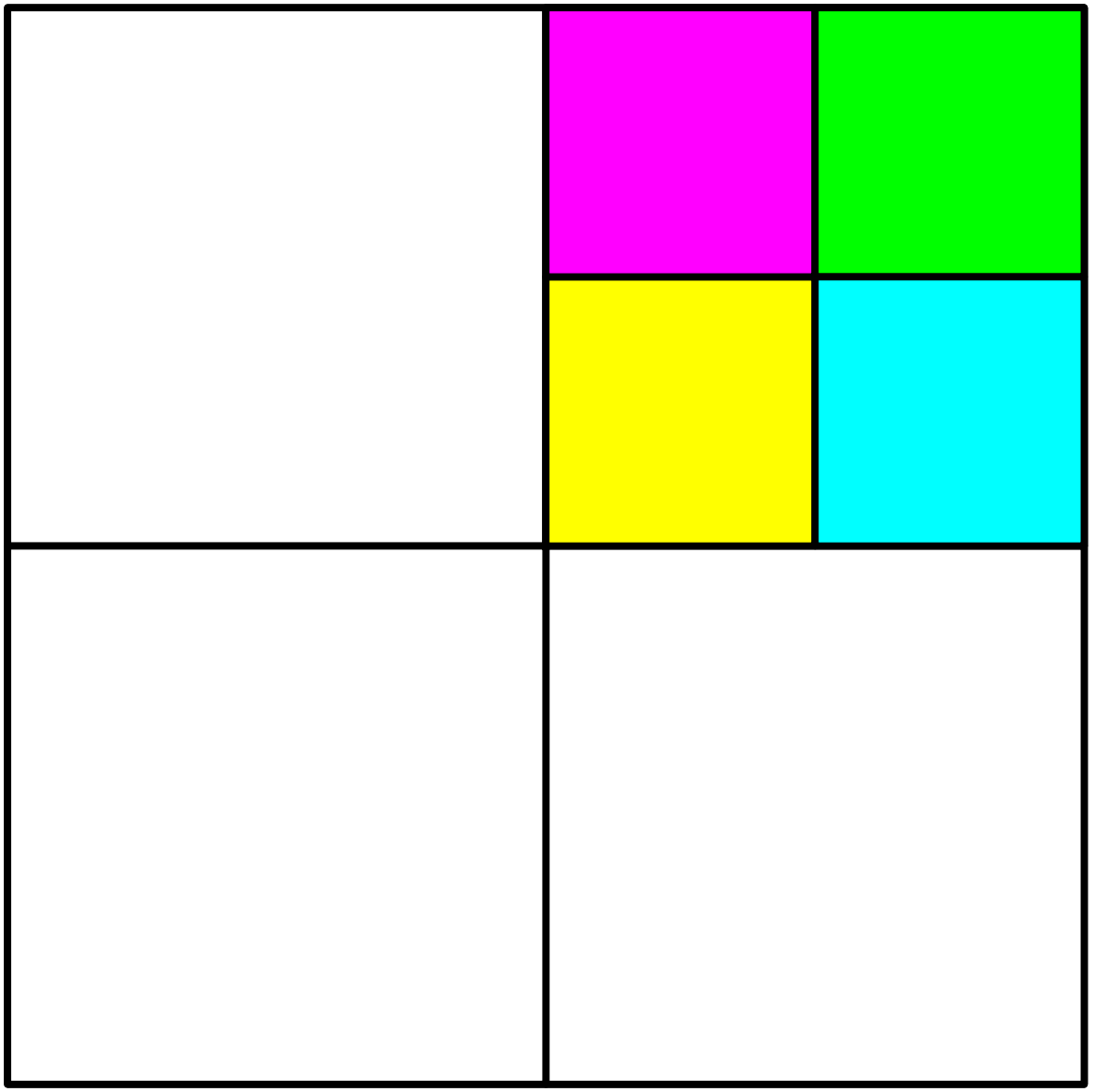
Subdividing a Square



For what values of n can a square be subdivided into n squares?

1 ~~2~~ ~~3~~ 4 ~~5~~ 6 7 8 9 10 11 12

The Key Insight



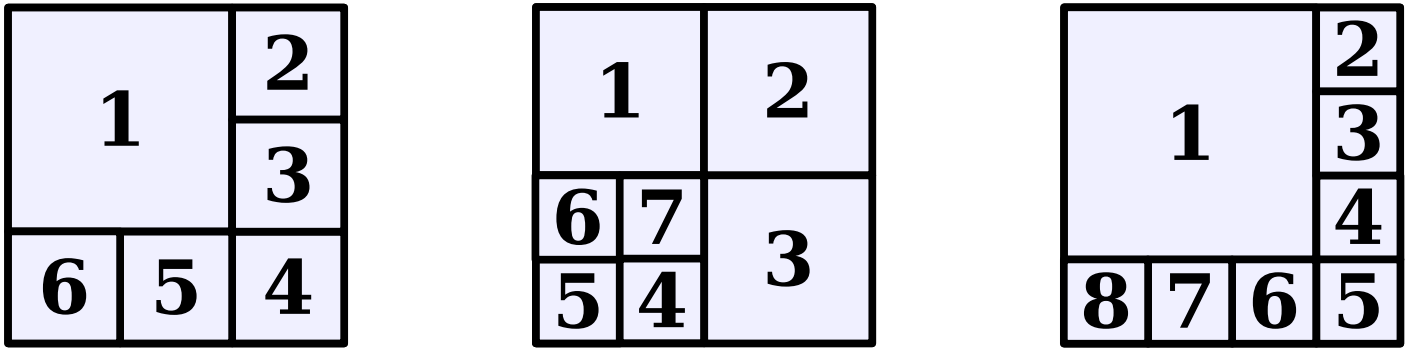
The Key Insight

- If we can subdivide a square into n squares, we can also subdivide it into $n + 3$ squares.
- Since we can subdivide a bigger square into 6, 7, and 8 squares, we can subdivide a square into n squares for any $n \geq 6$:
  - For multiples of three, start with 6 and keep adding three squares until n is reached.
  - For numbers congruent to one modulo three, start with 7 and keep adding three squares until n is reached.
  - For numbers congruent to two modulo three, start with 8 and keep adding three squares until n is reached.

Theorem: For any $n \geq 6$, it is possible to subdivide a square into n smaller squares.

Proof: Let $P(n)$ be the statement "a square can be subdivided into n smaller squares." We will prove by induction that $P(n)$ holds for all $n \geq 6$, from which the theorem follows.

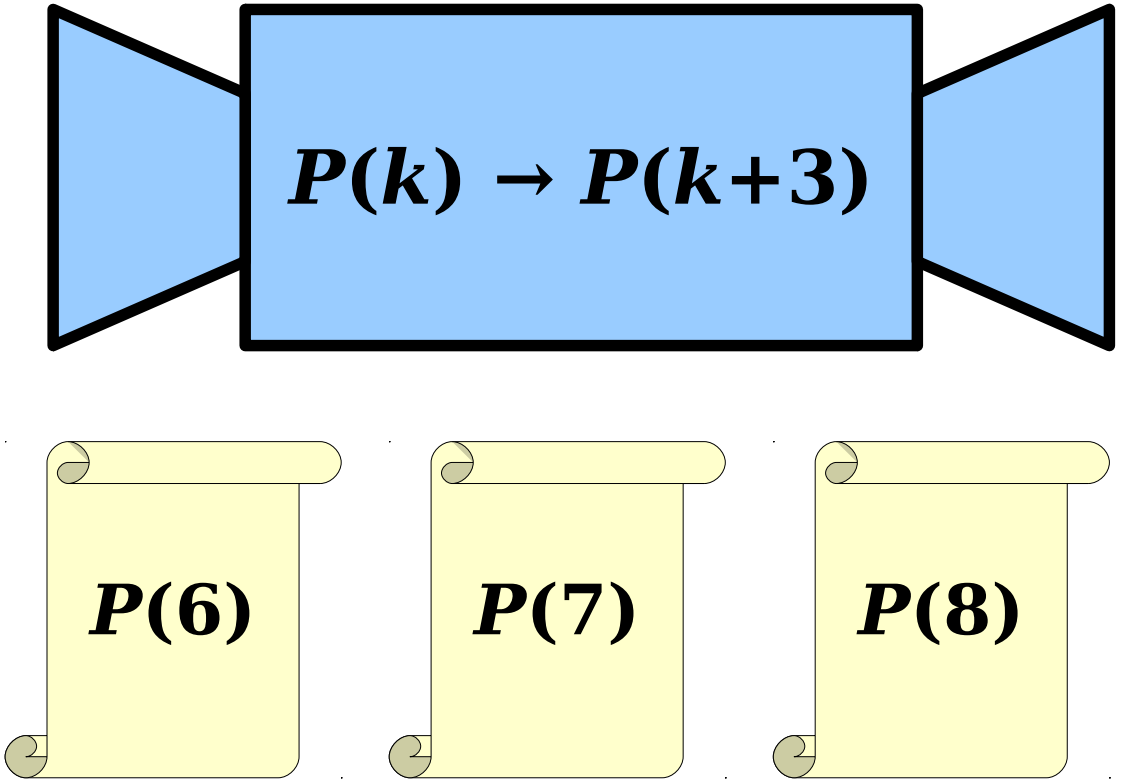
As our base cases, we prove $P(6)$, $P(7)$, and $P(8)$, that a square can be subdivided into 6, 7, and 8 squares. This is shown here:



For the inductive step, assume that for some arbitrary $k \geq 6$ that $P(k)$ is true and that a square can be subdivided into k squares. We prove $P(k+3)$, that a square can be subdivided into $k+3$ squares. To see this, start by obtaining (via the inductive hypothesis) a subdivision of a square into k squares. Then, choose any of the squares and split it into four equal squares. This removes one of the k squares and adds four more, so there will be a net total of $k+3$ squares. Thus $P(k+3)$ holds, completing the induction. ■

Why This Works

- This induction has three consecutive base cases and takes steps of size three.
- Thinking back to our “induction machine” analogy:



Generalizing Induction

- When doing a proof by induction,
  - feel free to use multiple base cases, and
  - feel free to take steps of sizes other than one.
- Just be careful to make sure you cover all the numbers you think that you're covering!
  - We won't require that you prove you've covered everything, but it doesn't hurt to double-check!

More on Square Subdivisions

- There are a ton of interesting questions that come up when trying to subdivide a rectangle or square into smaller squares.
- In fact, one of the major players in early graph theory (William Tutte) got his start playing around with these problems.
- Good starting resource: this Numberphile video on ***Squaring the Square***.

Time-Out for Announcements!

CS+SOCIAL GOOD

WINTER **MIXER**

MONDAY, FEBRUARY 12 **5-6 PM**

OLD UNION **200**

Come meet fellow students and teachers who are passionate about using technology for good!

Everyone is welcome, and there will be free boba and snacks!

Problem Set Five

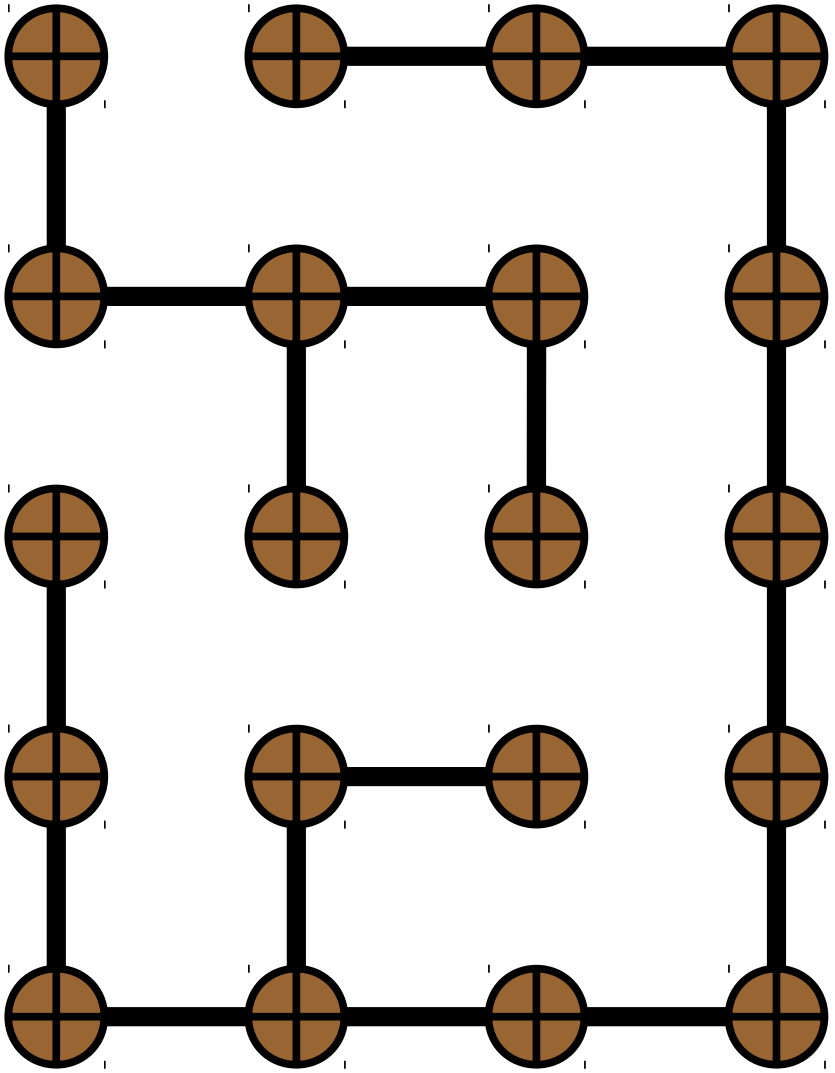
- Problem Set Four was due at 2:30PM today.
- Problem Set Five goes out today. It's due next Friday at 2:30PM.
  - Play around with everything we've covered so far, plus a healthy dose of induction and inductive problem-solving.
  - There is no checkpoint problem, and there are no checkpoints from here out out.

Back to CS103!

A Motivating Question: ***Rat Mazes***

Rat Mazes

- Suppose you want to make a rat maze consisting of an $n \times m$ grid of pegs with slats between them.
- The maze should have these properties:
  - There is one entrance and one exit in the border.
  - Every spot in the maze is reachable from every other spot.
  - There is exactly one path from each spot in the maze to each other spot.

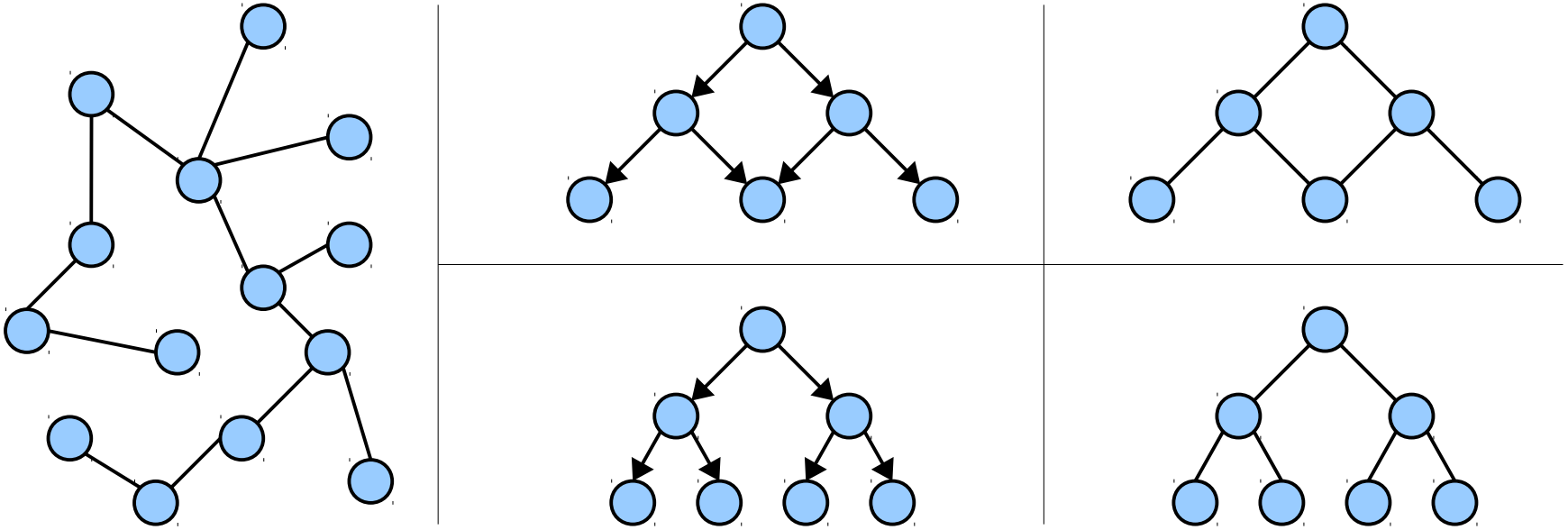


Question: If you have an $n \times m$ grid of pegs, how many slats do you need to make?

A Special Type of Graph: ***Trees***

- A ***tree*** is a connected, nonempty graph with no simple cycles.

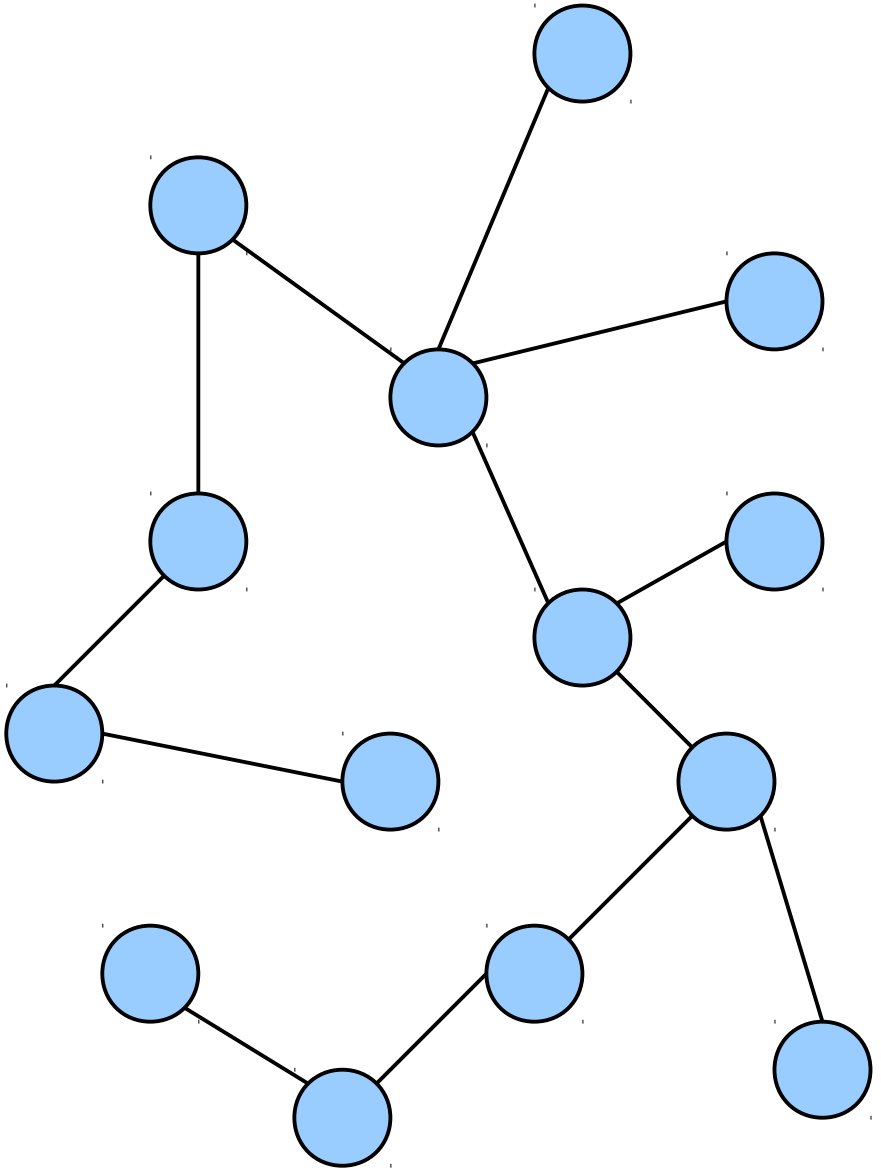
According to the above definition of trees, how many of these graphs are trees?



Answer at **PollEv.com/cs103** or
text **CS103** to **22333** once to join, then a number.

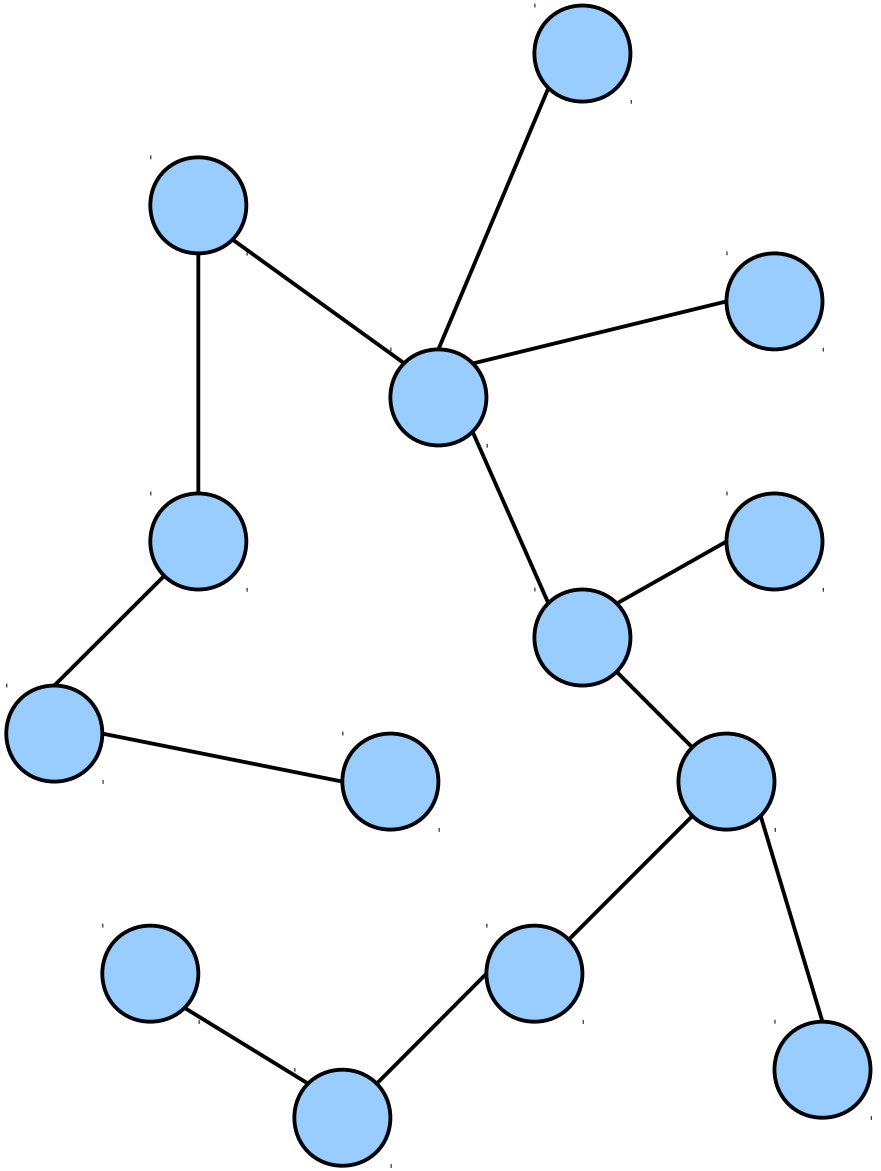
Trees

- A ***tree*** is a connected, nonempty graph with no simple cycles.
- Trees have tons of nice properties:
  - They're ***maximally acyclic*** (adding any missing edge creates a simple cycle)
  - They're ***minimally connected*** (deleting any edge disconnects the graph)
- Proofs of these results are in the course reader if you're interested. They're also great exercises.



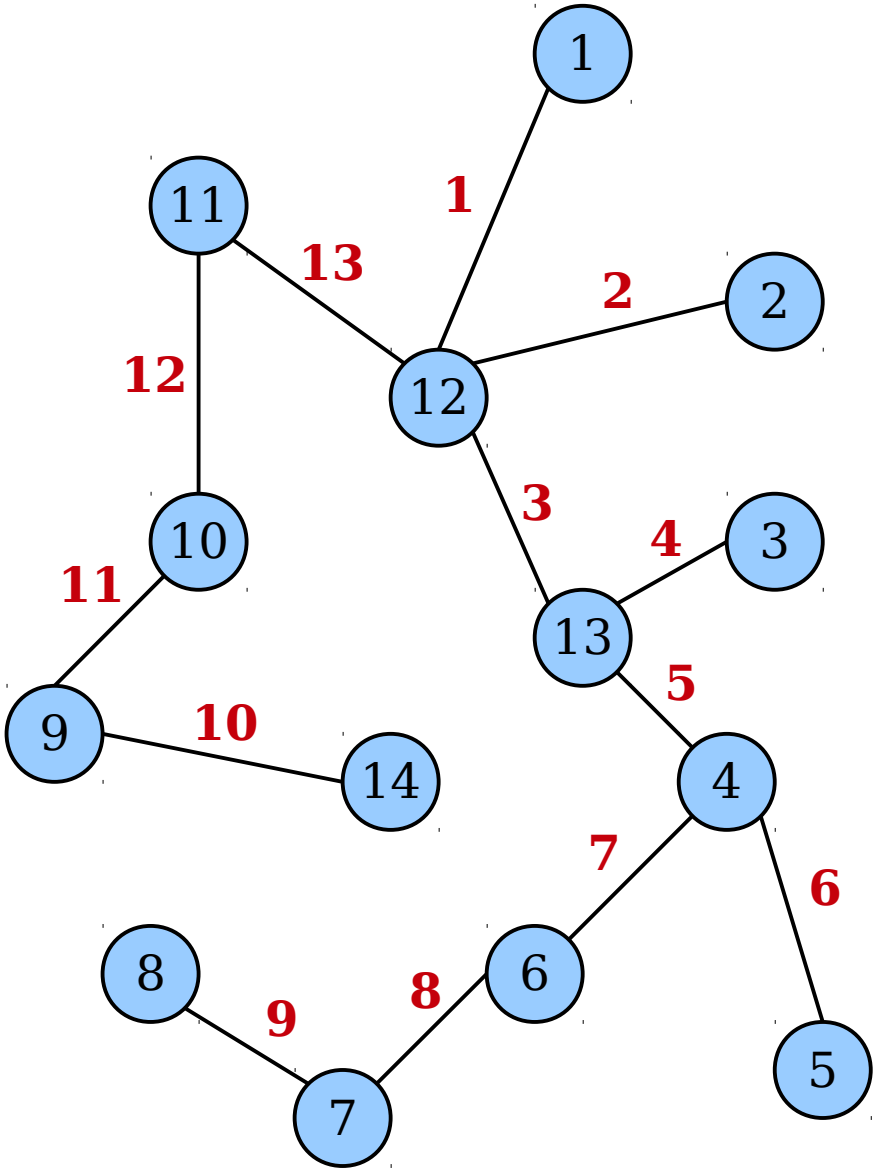
Trees

- ***Theorem:*** If T is a tree with at least two nodes, then deleting any edge from T splits T into two nonempty trees T_1 and T_2.
- ***Proof:*** Left as an exercise to the reader. ☺



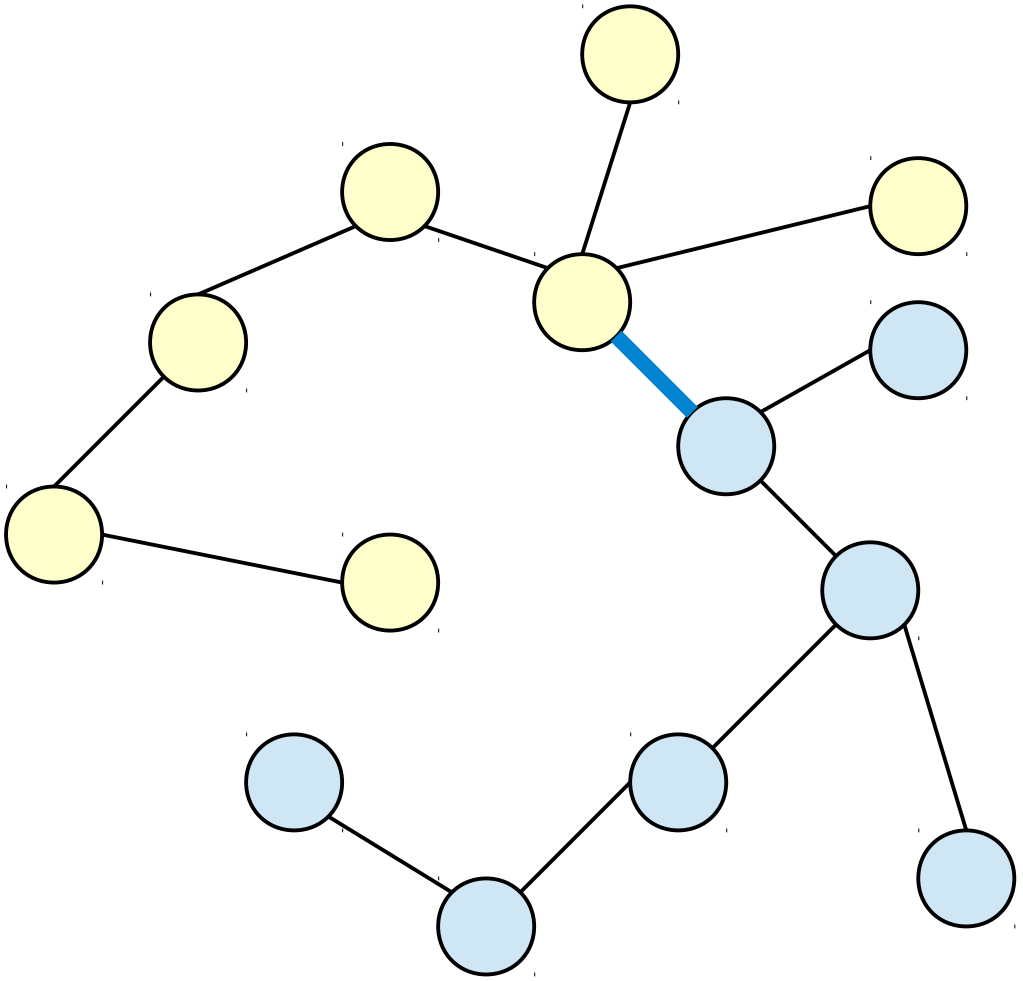
Trees

- ***Theorem:*** If T is a tree with $n \geq 1$ nodes, then T has exactly n-1 edges.
- ***Proof:*** Up next!



Our Base Case

Assume any tree with at most k nodes has one more node than edge.



Consider an arbitrary tree with $k+1$ nodes.

Suppose there are r nodes in the yellow tree.

Then there are $(k+1)-r$ nodes in the blue tree.

There are $r-1$ edges in the yellow tree and $k-r$ edges in the blue tree.

Adding in the initial edge we cut, there are $r-1 + k-r + 1 = k$ edges in the original tree.

Theorem: If T is a tree with $n \geq 1$ nodes, then T has n-1 edges.

Proof: Let $P(n)$ be the statement "any tree with n nodes has n-1 edges." We will prove by induction that $P(n)$ holds for all $n \geq 1$, from which the theorem follows.

As a base case, we will prove $P(1)$, that any tree with 1 node has 0 edges. Any such tree has single node, so it cannot have any edges.

Now, assume for some arbitrary $k \geq 1$ that $P(1)$, $P(2)$, …, and $P(k)$ are true, so any tree with between 1 and k nodes has one more node than edge. We will prove $P(k+1)$, that any tree with $k+1$ nodes has k edges.

Consider any tree T with $k+1$ nodes. Since T has at least two nodes and is connected, it must contain at least one edge. Choose any edge in T and delete it. This splits T into two nonempty trees T_1 and T_2. Every edge in T is part of T_1, is part of T_2, or is the initial edge we deleted.

Let r be the number of nodes in T_1. Since every node in T belongs to either T_1 or T_2, we see that T_2 has $(k+1)$-r nodes. Additionally, since T_1 and T_2 are nonempty, neither T_1 nor T_2 contains all the nodes from T. Therefore, T_1 and T_2 each have between 1 and k nodes. We can then apply our inductive hypothesis to see that T_1 has r-1 edges and T_2 has k-r edges. Thus the total number of edges in T is $1 + (r\text{-}1) + (k\text{-}r) = k$, as required. Therefore, $P(k+1)$ is true, completing the induction. ■

Theorem: If T is a tree with $n \geq 1$ nodes, then T has n-1 edges.

Proof: Let $P(n)$ be the statement "any tree with n nodes has n-1 edges." We will prove by induction that $P(n)$ holds for all $n \geq 1$, from which the theorem follows.

As a base case, we will prove $P(1)$, that any tree with 1 node has 0 edges. Any such tree has single node, so it cannot have any edges.

Now, assume for some arbitrary $k \geq 1$ that $P(1)$, $P(2)$, ..., and $P(k)$ are true, so any tree with between 1 and k nodes has one more node than edge. We will prove $P(k+1)$, that any tree with $k+1$ nodes has k edges.

Which of the following best describes the structure of the inductive step in this proof?

A. Assume $P(1)$, then prove $P(k+1)$.
B. Assume $P(k)$, then prove $P(k+1)$.
C. Assume $P(1)$, then prove $P(1)$, ..., $P(k)$, and $P(k+1)$.
D. Assume $P(1)$, ..., and $P(k)$, then prove $P(k+1)$.
E. None of these, or more than one of these.

Answer at **PollEv.com/cs103** or text **CS103** to **22333** once to join, then ***A***, ..., or ***E***.

T_1

Therefore, T_1 and T_2 each have between 1 and k nodes. We can then apply our inductive hypothesis to see that T_1 has r-1 edges and T_2 has k-r edg $r) = k$, as req n. ■

Complete Induction

- If the following are true:
  - $P(0)$ is true, and
  - If $P(0)$, $P(1)$, $P(2)$, …, $P(k)$ are true, then $P(k+1)$ is true as well.

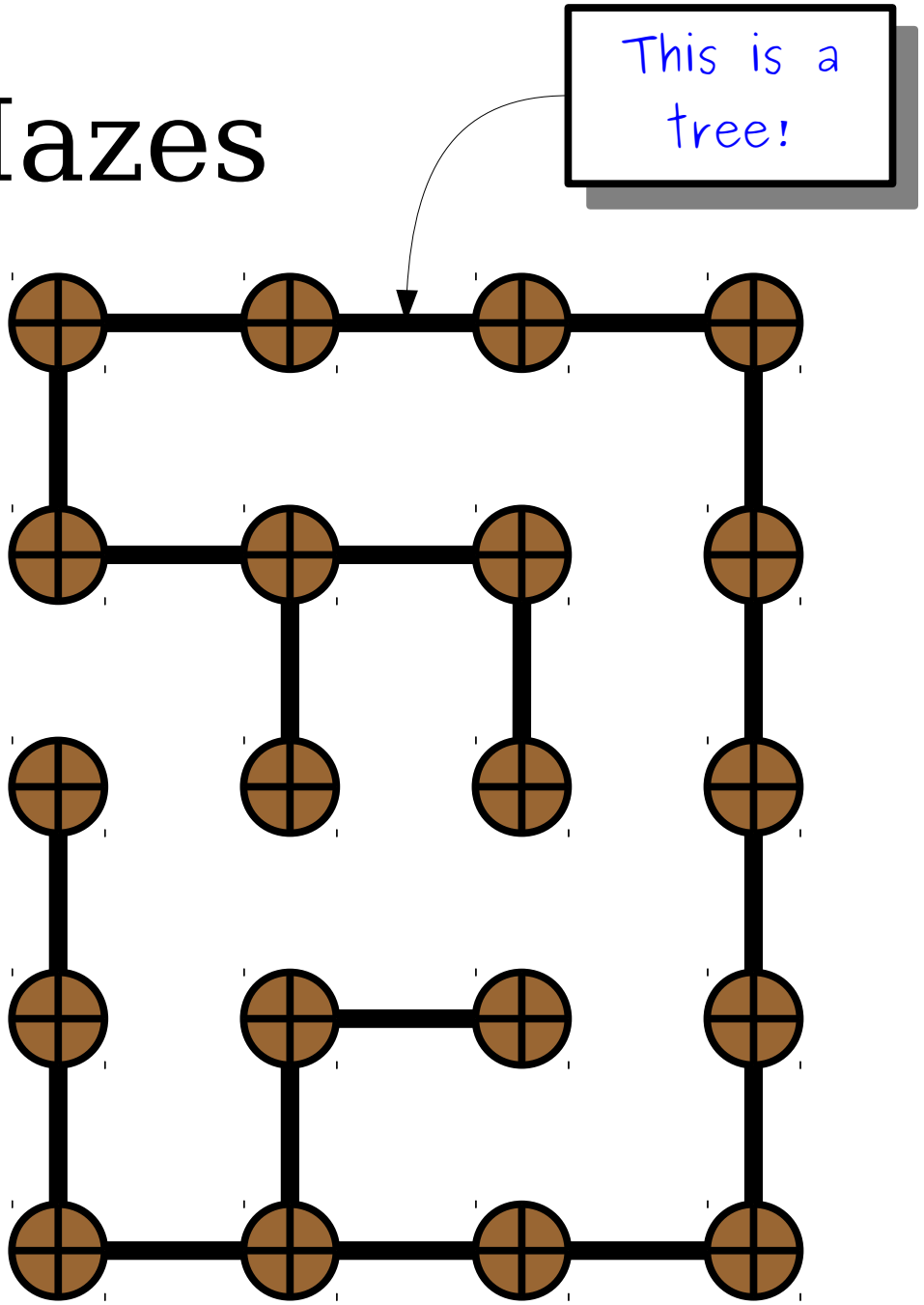
  then $P(n)$ is true for all $n \in \mathbb{N}$.
- This is called the ***principle of complete induction*** or the ***principle of strong induction***.
  - (This also works starting from a number other than 0; just modify what you're assuming appropriately.)

When Use Complete Induction?

- Normal induction is good for when you are shrinking the problem size by exactly one.
  - Peeling one final term off a sum.
  - Making one weighing on a scale.
  - Considering one more action on a string.
- Complete induction is good when you are shrinking the problem, but you can't be sure by how much.
  - In the previous example, if we delete a random edge, we can't know in advance how big the resulting trees will be.

Rat Mazes

- Suppose you want to make a rat maze consisting of an $n \times m$ grid of pegs with slats between them.
- ***Question:*** How many slats do you need to create?
- ***Answer:*** $mn - 2$.



For more on trees, take CS161 / 261 / 267!

An Important Milestone

Recap: ***Discrete Mathematics***

- The past five weeks have focused exclusively on discrete mathematics:

Induction	Functions
Graphs	The Pigeonhole Principle
Relations	Mathematical Logic
Set Theory	Cardinality

- These are building blocks we will use throughout the rest of the quarter.
- These are building blocks you will use throughout the rest of your CS career.

Next Up: ***Computability Theory***

- It's time to switch gears and address the limits of what can be computed.
- We'll explore these questions:
  - How do we model computation itself?
  - What exactly is a computing device?
  - What problems can be solved by computers?
  - What problems *can't* be solved by computers?
- ***Get ready to explore the boundaries of what computers could ever be made to do.***

Next Time

- ***Formal Language Theory***
  - How are we going to formally model computation?
- ***Finite Automata***
  - A simple but powerful computing device made entirely of math!
- ***DFAs***
  - A fundamental building block in computing.