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 property. 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

- 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.
- ...

Proof by Induction

- A ***proof by induction*** is a way to use 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 Summations

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

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

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

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

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

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 property 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.

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 $k \in \mathbb{N}$ 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 $k \in \mathbb{N}$ 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 stating $P(k+1)$, which is what we want to prove. Now, we can use any proof technique we want to try to prove it.

Theorem: 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… $2^0 - 1$ 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 $k \in \mathbb{N}$ 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 $k \in \mathbb{N}$ that $P(k)$ holds, meaning that

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

We n … um
of th…
notic…

$$2^{…} \qquad \text{(via (1))}$$
$$= 2(2^k) - 1$$
$$= 2^{k+1} - 1.$$

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.

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

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 $k \in \mathbb{N}$ 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 later today.

Structuring a Proof by Induction

- Define some property P that you'll show, by induction, is true for all natural numbers.
- Prove the base case:
  - State that you're going to prove the property holds for 0, then go prove it.
- Prove the inductive step:
  - Say that you're assuming P is true for some natural number k, then write out exactly what that means.
  - Say that you're going to prove P is true for $k+1$, then write out exactly what that means.
  - Prove that P is true for $k+1$ 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.

Induction, Intuitively

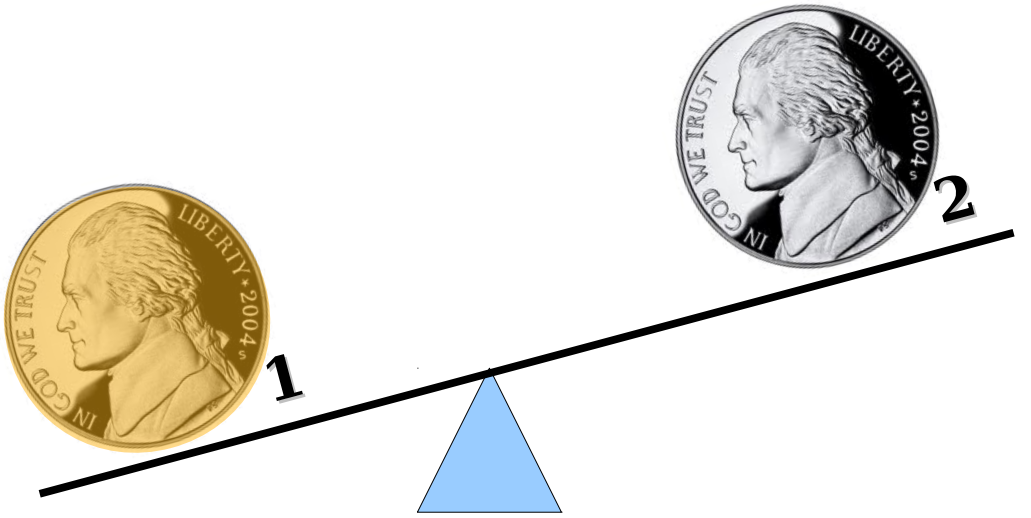
- You can imagine an “machine” that turns proofs that the property holds for k into proofs that the property holds for $k + 1$.
- Starting with a proof that the property holds for 0, we can run the machine as many times as we'd like to get proofs for 1, 2, 3, … .
- The principle of mathematical induction says that this style of reasoning is a rigorous argument.

The Counterfeit Coin Problem

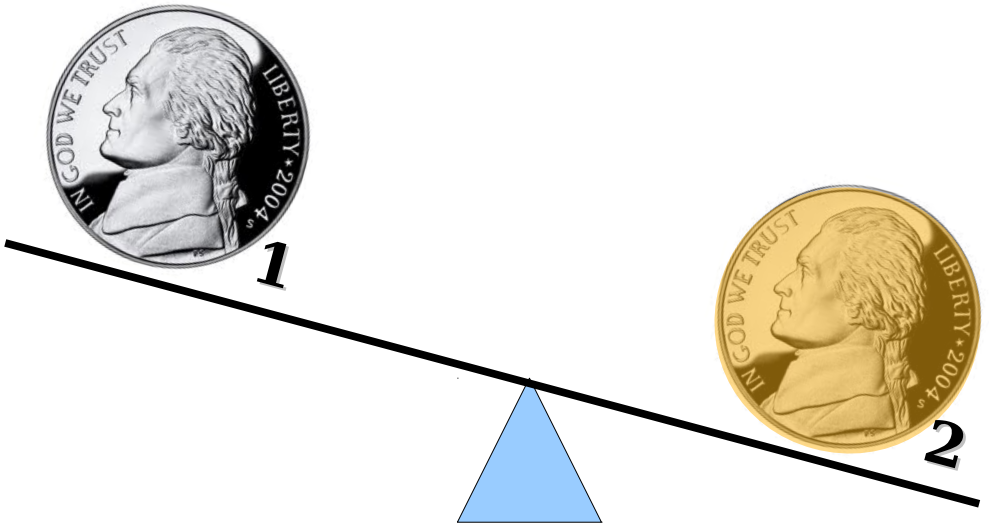
Problem Statement

- 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.

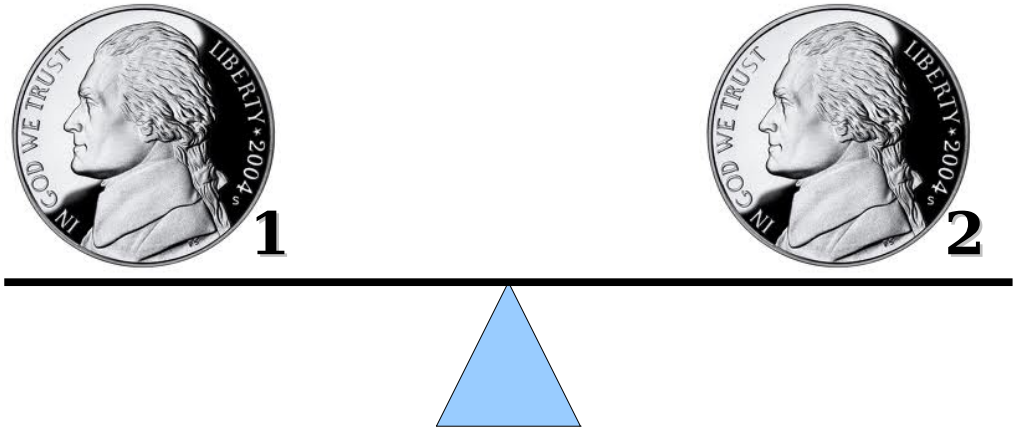
Finding the Counterfeit Coin



Finding the Counterfeit Coin



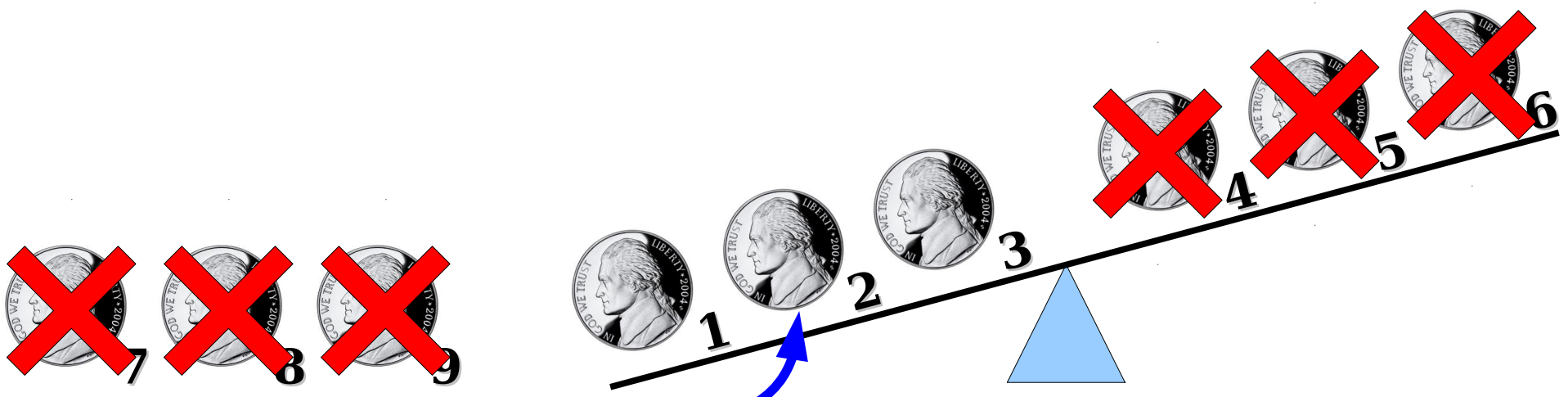
Finding the Counterfeit Coin



A Harder Problem

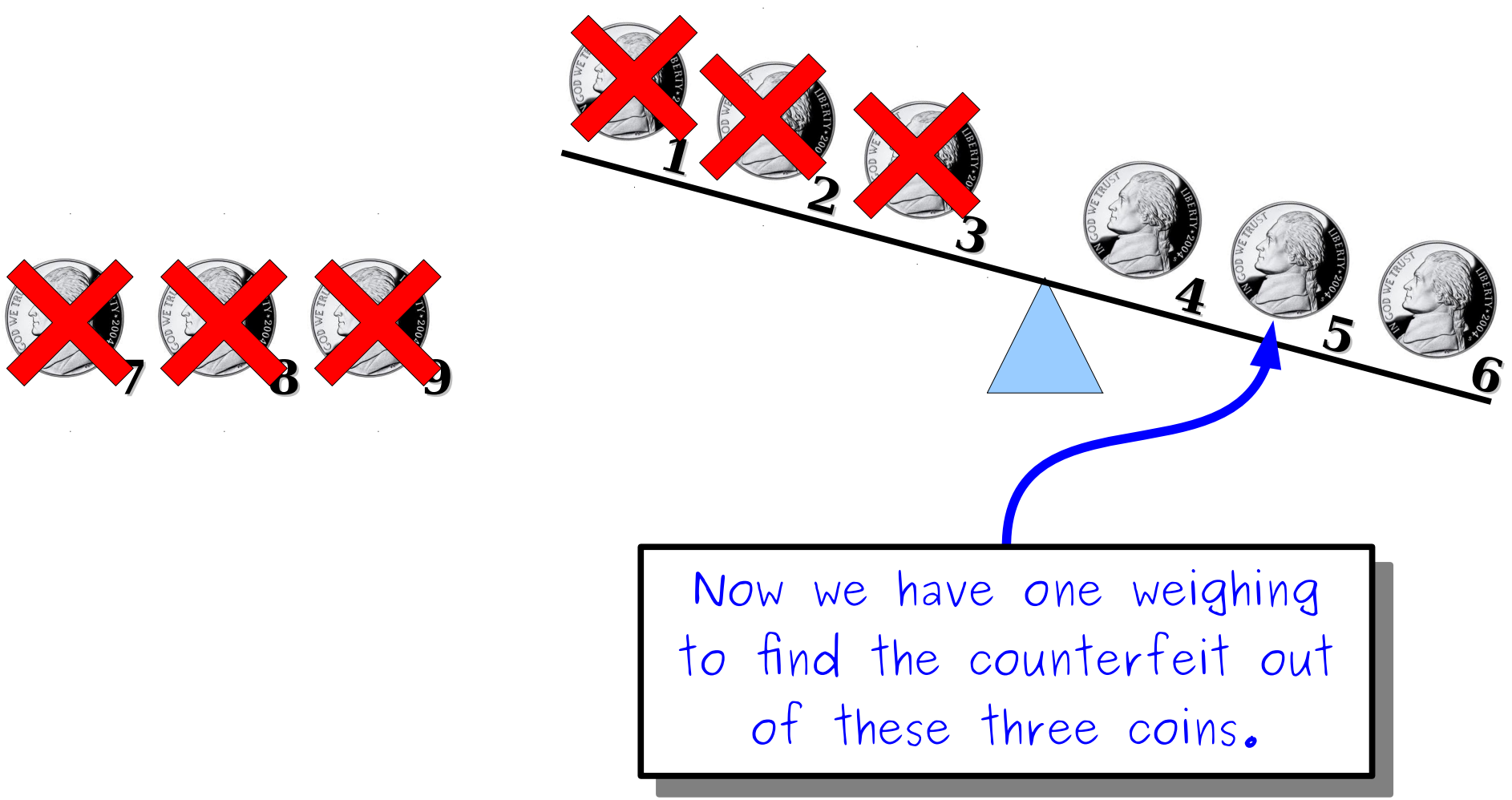
- You are given a set of ***nine*** seemingly identical coins, eight of which are real and one of which is counterfeit.
- The counterfeit coin weighs more than the rest of the coins.
- You are given a balance. Using only ***two*** weighings on the balance, find the counterfeit coin.

Finding the Counterfeit Coin

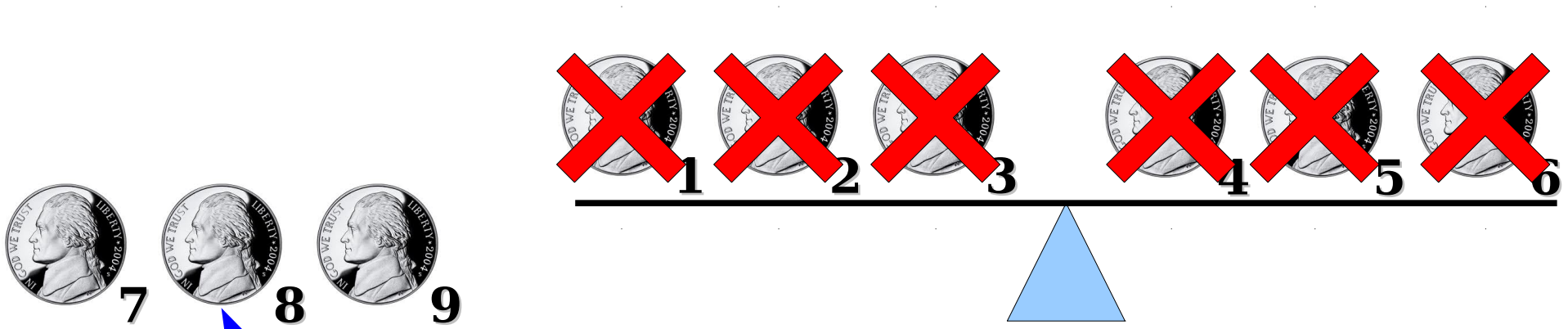


Now we have one weighing to find the counterfeit out of these three coins.

Finding the Counterfeit Coin



Finding the Counterfeit Coin



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

If we have n weighings on the scale, what is the largest number of coins out of which we can find the counterfeit?

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 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 property 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 that $P(k)$ is true for some $k \in \mathbb{N}$, 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 that $P(k)$ is true for some $k \in \mathbb{N}$, 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 that $P(k)$ is true for some $k \in \mathbb{N}$, 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 w…in, it's vac…ed.

For the…we can fin…hat we can…

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.

Suppos…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.

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 that $P(k)$ is true for some $k \in \mathbb{N}$, 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

- 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?

Time-Out for Announcements!

Problem Set Four

- Problem Set Four checkpoint was due at 3:00PM today. That's the last checkpoint of the quarter!
- The remaining problems are due on Friday at the start of class.
  - ***Make sure to start early!*** You've probably figured this out by now, but these questions take time to think over.
  - ***Ask questions when you have them!*** You can ask on Piazza or in office hours.

Midterm Exam

- The first midterm exam is next **Monday, February 8** from **7PM – 10PM**, location TBA.
- Covers material from PS1 – PS3. Later concepts will not be tested (yet).
- You're responsible for Lectures 00 – 08 and for topics covered on PS1 – PS3.
- Students with OAE accommodations: please contact us ***immediately*** if you need to take the exam at an alternate time or need extended time.

Midterm Exam

- ***We want you to do well on this exam***. We're not trying to weed out weak students. We're not trying to enforce a curve where there isn't one. We want you to show what you've learned up to this point so that you get a sense for where you stand and where you can improve.
- The purpose of this midterm is to give you a chance to show what you've learned in the past month. It is not designed to assess your “mathematical potential” or “innate mathematical ability.”

Midterm Exam

- The exam is
  - closed-book,
  - closed-computer, and
  - limited-note.
- You can have a single, double-sided sheet of 8.5” × 11” paper with notes with you while you take the exam.
- You may want to start thinking about what you're going to put on that note sheet.
- Our advice: ***write your own notes***. The act of creating them will help solidify your understanding.

Practice Midterm Exam

- To help you prepare for the midterm, we'll be holding a practice midterm exam on **Wednesday, 7PM – 10PM** in **Bishop Auditorium**.
- We've written two exams – a practice exam and a real exam – that have similar structure and style. We'll give you the practice midterm on Wednesday under realistic conditions so that you can prepare for the exam.
- The TAs and I will be on hand to answer your questions.
- Can't make it? We'll release the practice exam online and the solutions in hardcopy.

Extra Practice Problems

- We'll be releasing three sets of cumulative review problems this week that you can use to prepare for the exam.
- We strongly recommend working through these practice problems. They're a great way to get additional practice with the material and to see where you need to study.
- Solutions will be released in class and will be available for pickup in the Gates filing cabinet.

Advice from Generations Past

- We've released a handout (Handout 19) containing advice from previous CS103 students about how to do well on the exam.
- Read it over. There's some good advice in there!

Your Questions

“Keith, how can I get better at maths?
(specifically CS103 and CS109)”

Practice, practice, practice! Get in as much practice as you can. Do the extra practice problems. Ask the TAs to review your work honestly but politely. Identify what skills you need to work on, then focus down those areas.

"What were some of your favorite classes when you were an undergrad?"

I really liked my PWR section from my freshman year (I am a much better writer for the experience). I also enjoyed "A History of Russian Music" (culture!) and "Planetary Exploration" (space!)

On the CS side, I loved CS143, CS154, CS161/261/361, and CS140. Great classes. Highly recommended!

Back to CS103!

How Not To Induct

Something's Wrong...

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 $k \in \mathbb{N}$ 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 $k \in \mathbb{N}$ 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

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

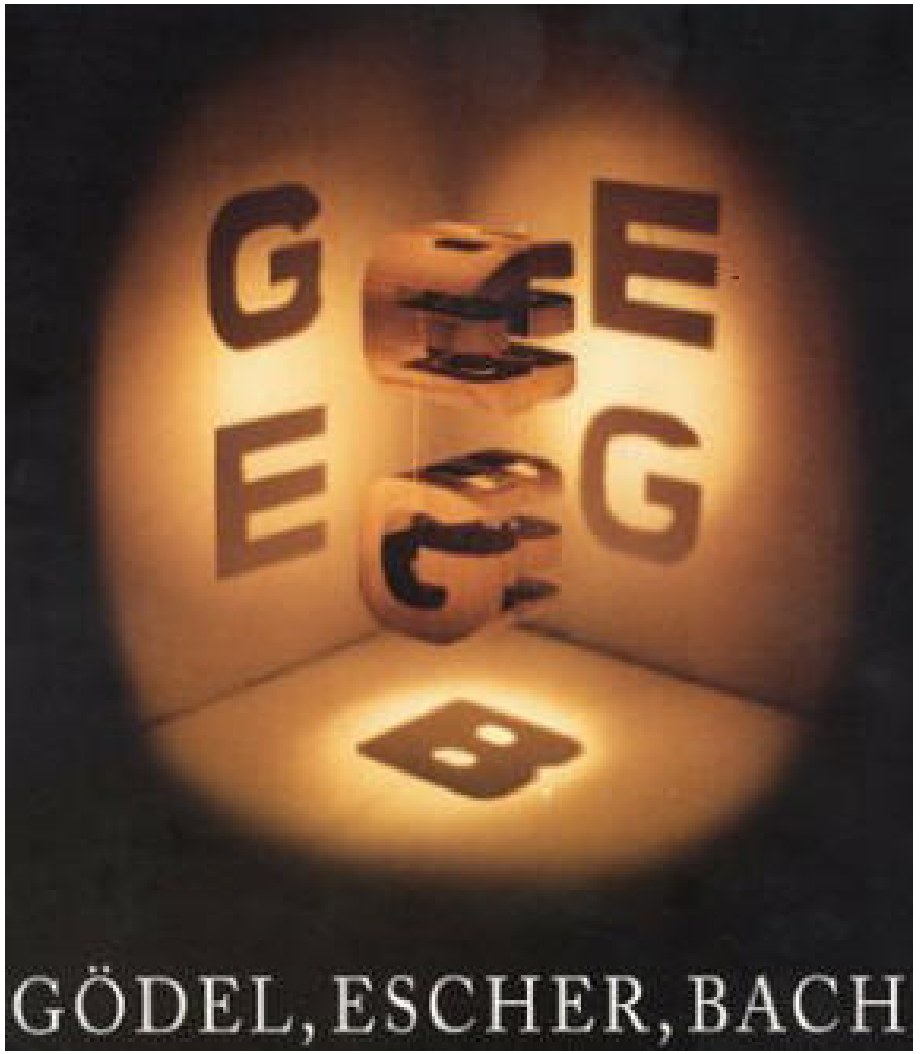
$$= 2^k + 2^k \qquad \text{(via (1))}$$

Therefore, $P(k + 1)$

You can prove **anything** from a faulty assumption. This is called the **principle of explosion.**

The MU Puzzle

Gödel, Escher Bach: An Eternal Golden Braid



- Pulitzer-Prize winning book exploring recursion, computability, and consciousness.
- Written by Douglas Hofstadter, cognitive scientist at Indiana University.
- A great (but dense!) read.

The MU Puzzle

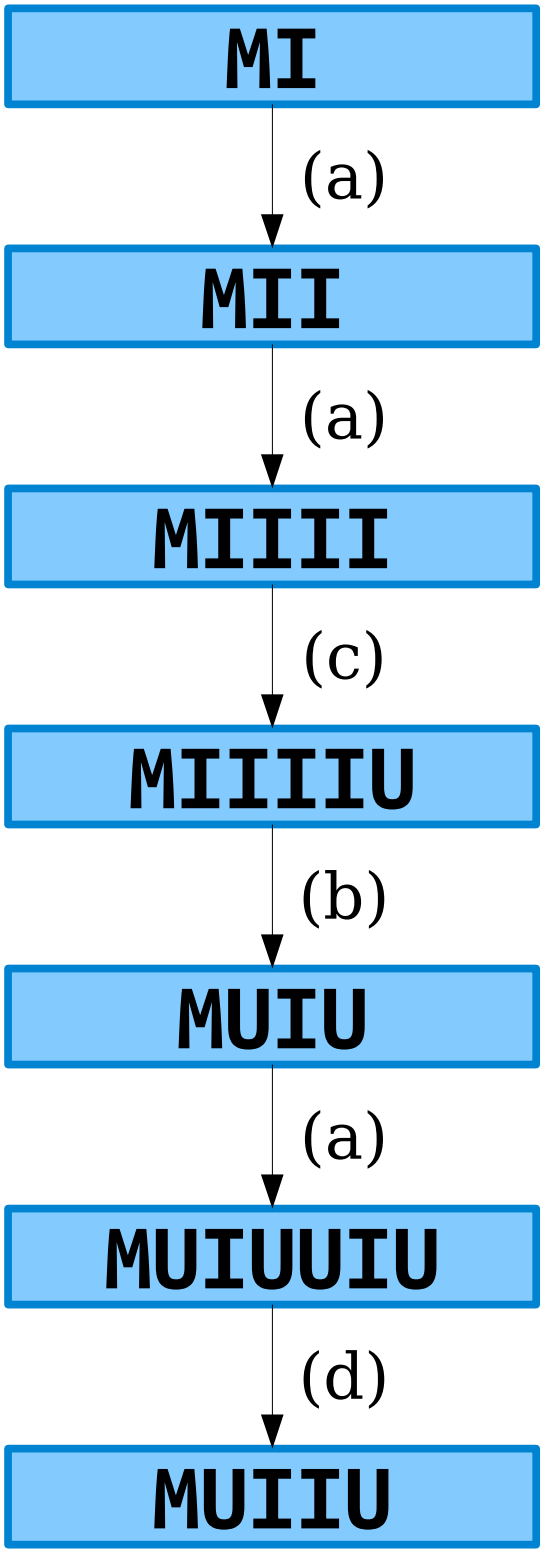
- Begin with the string `MI`.
- Repeatedly apply one of the following operations:
  - Double the contents of the string after the `M`: for example, `MIIU` becomes `MIIUIIU`, or `MI` becomes `MII`.
  - Replace `III` with `U`: `MIIII` becomes `MUI` or `MIU`.
  - Append `U` to the string if it ends in `I`: `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 $k \in \mathbb{N}$. 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!

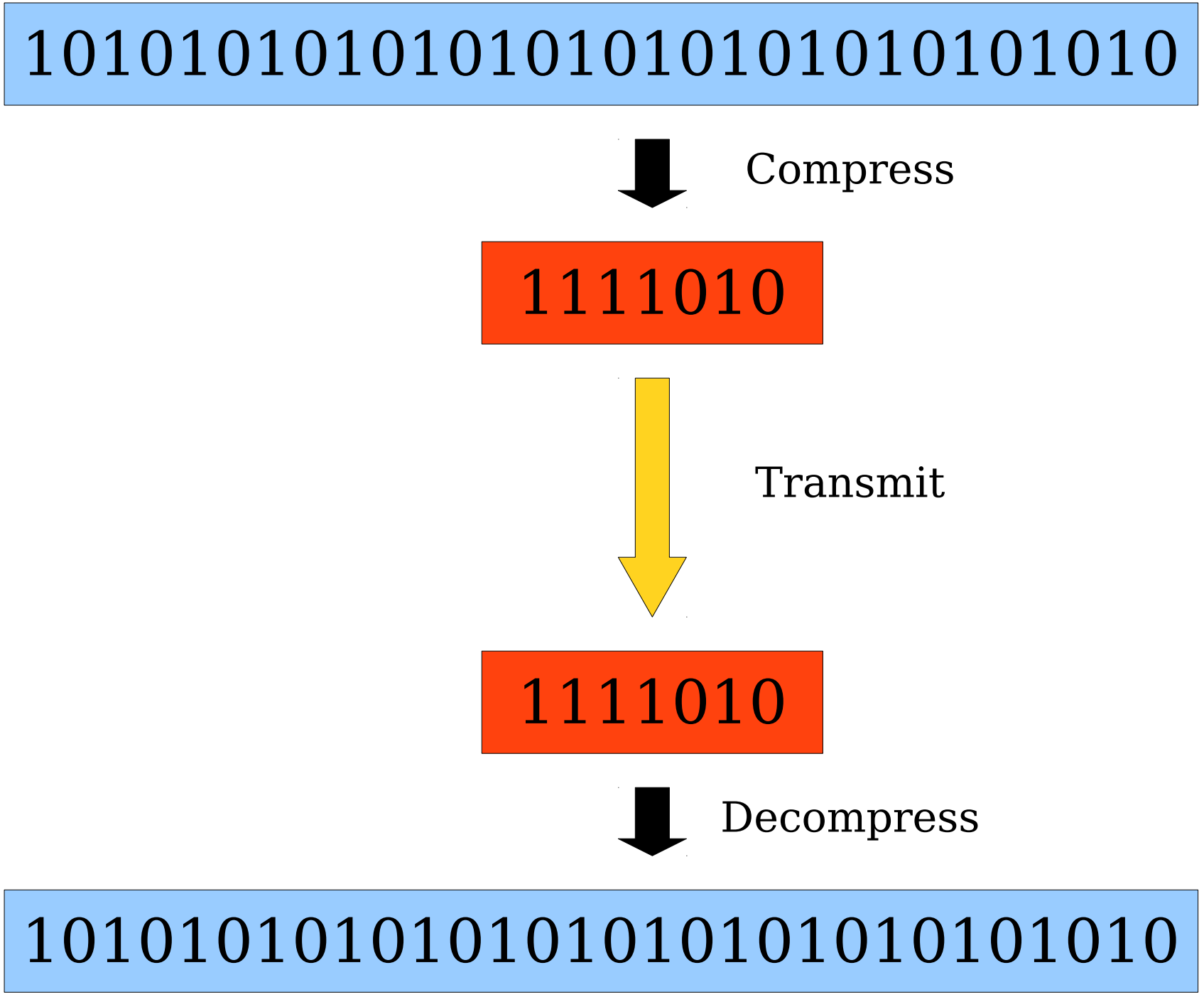
The Limits of Data Compression

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 (across a network, on DVDs, on a flash drive, etc.), 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.
- 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!

Next Time

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