Mathematical Induction

Part Two

Recap from Last Time

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

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

If it starts true…

…and it stays true…

and

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

then

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

…then it's always 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 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

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

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 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. ■

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. ■

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. ■

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 we have

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

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:

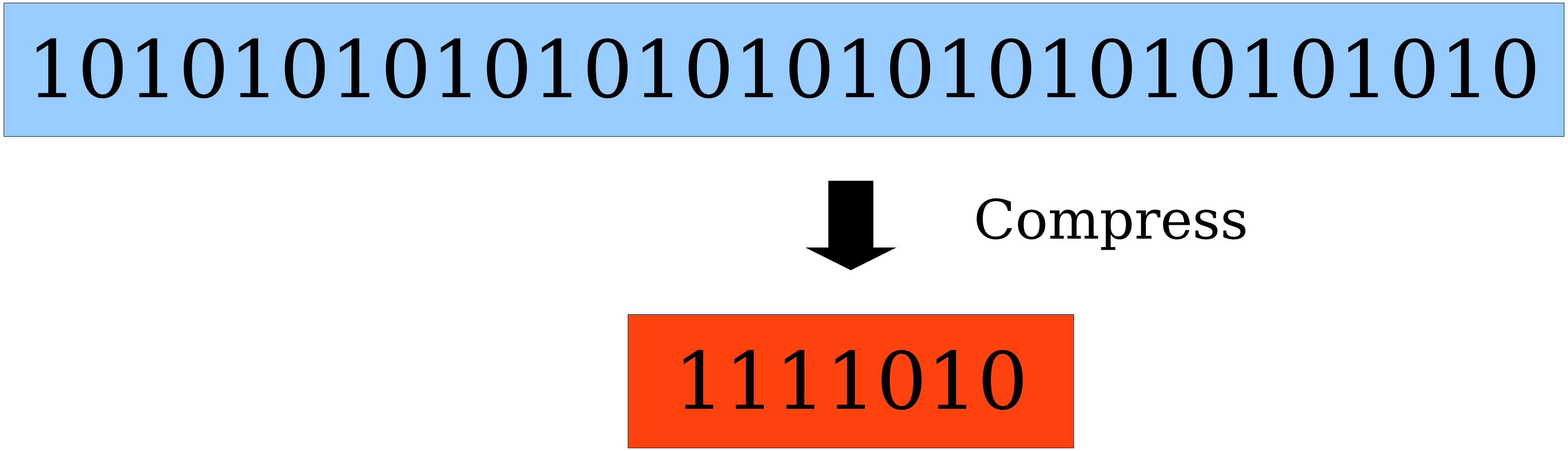
- 11011100
- 010101010101
- 0000
- ε (the ***empty string***)

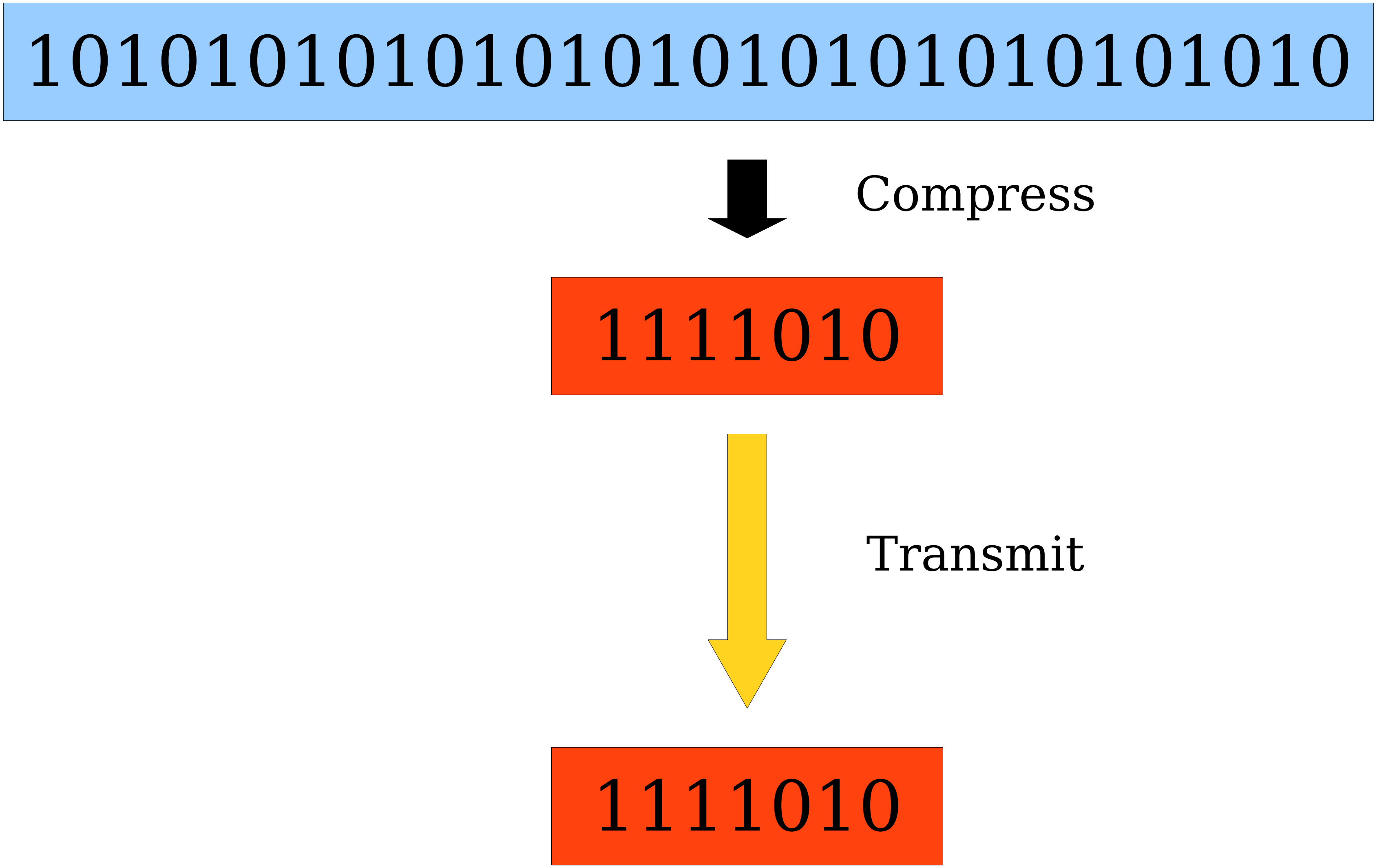
There are 2^n bitstrings of length n.

Data Compression

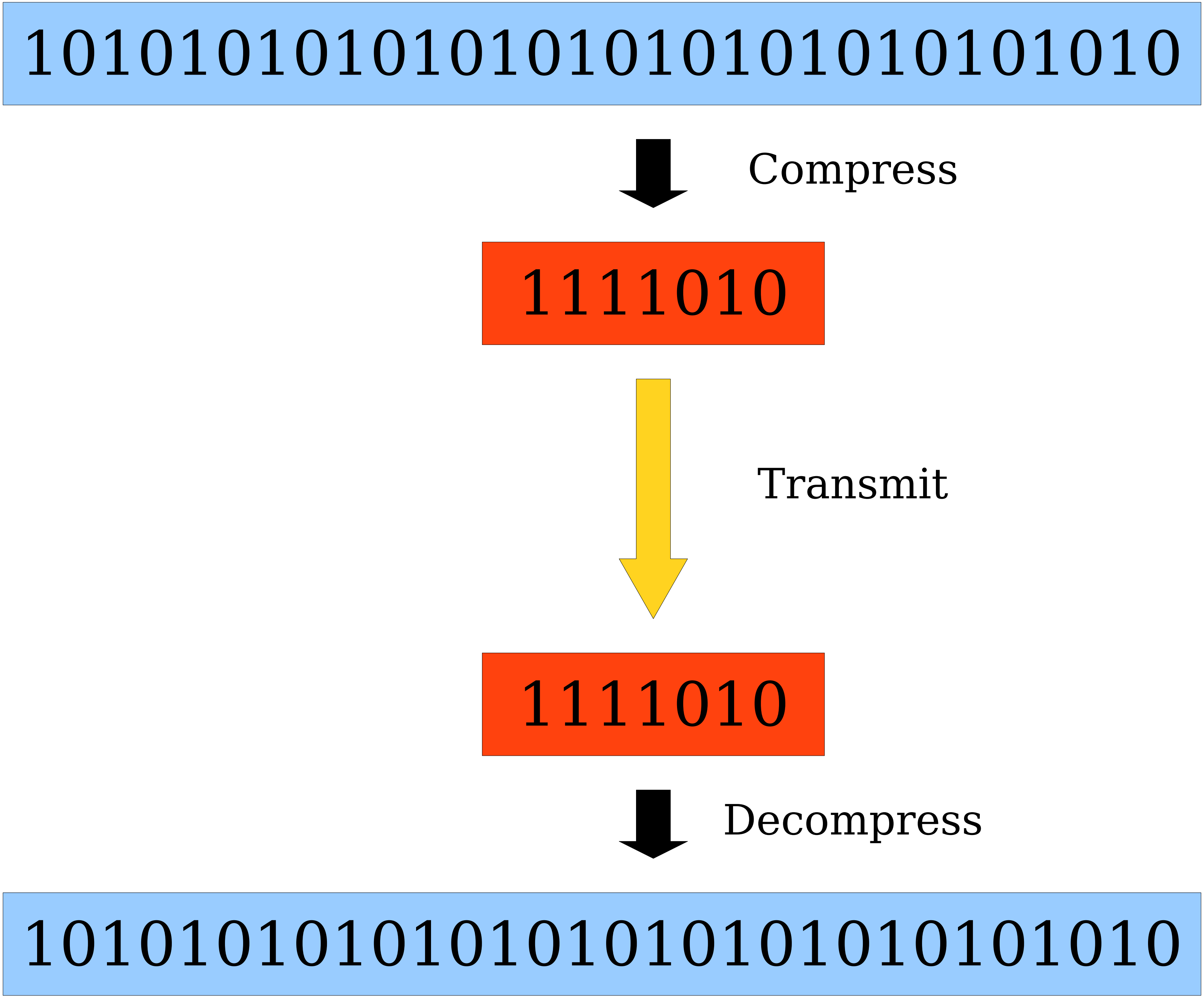
- Inside a computer, all data are represented as sequences of 0s and 1s (bitstrings)
- To transfer data over a network (or on a flash drive, if you're still into that), it is useful to reduce the number of 0s and 1s before transferring it.
- Most real-world data can be compressed by exploiting redundancies.
- 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***.

101010101010101010101010101010


101010101010101010101010101010
Compress
1111010



101010101010101010101010101010
Compress
1111010
Transmit
1111010



101010101010101010101010101010
Compress
1111010
Transmit
1111010
Decompress
101010101010101010101010101010

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!

All Horses are the Same Color

$P(n)$ = "All groups of n horses always have the same color"

All Horses are the Same Color

$P(0)$ = "All groups of 0 horses always have the same color"

Vacuously true!

Base case: $n = 0$

All Horses are the Same Color

Assume $P(k)$ = "All groups of k horses always have the same color"

Inductive hypothesis: $n = k$

All Horses are the Same Color

Prove $P(k+1)$ = "All groups of $k+1$ horses always have the same color"

Inductive hypothesis: $n = k+1$

All Horses are the Same Color

Prove $P(k+1)$ = "All groups of $k+1$ horses always have the same color"

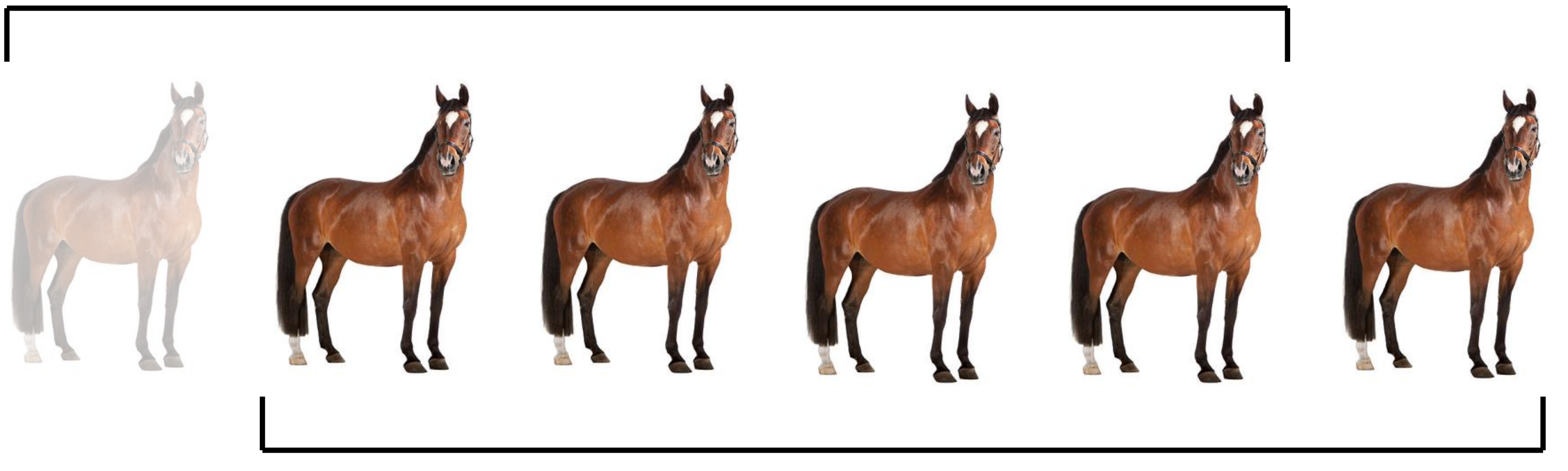
By $P(k)$, these k horses have the same color

Inductive hypothesis: $n = k+1$

All Horses are the Same Color

Prove $P(k+1)$ = "All groups of $k+1$ horses always have the same color"

By $P(k)$, these k horses have the same color



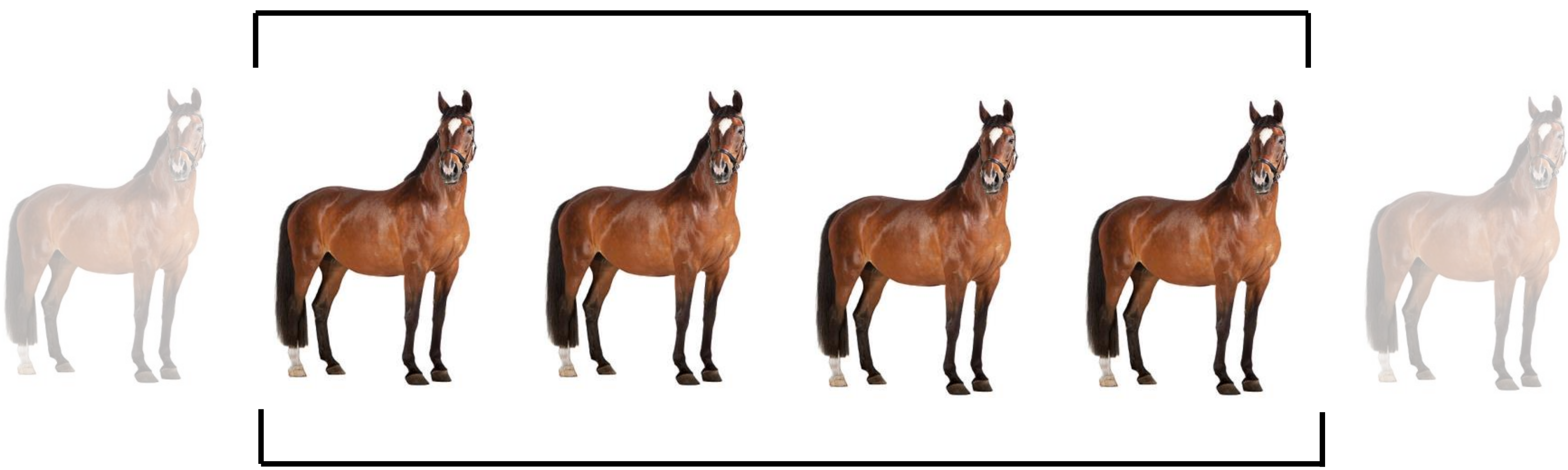
By $P(k)$, these k horses have the same color

Inductive hypothesis: $n = k+1$

All Horses are the Same Color

Prove $P(k+1)$ = "All groups of $k+1$ horses always have the same color"

These horses in the middle were in both sets

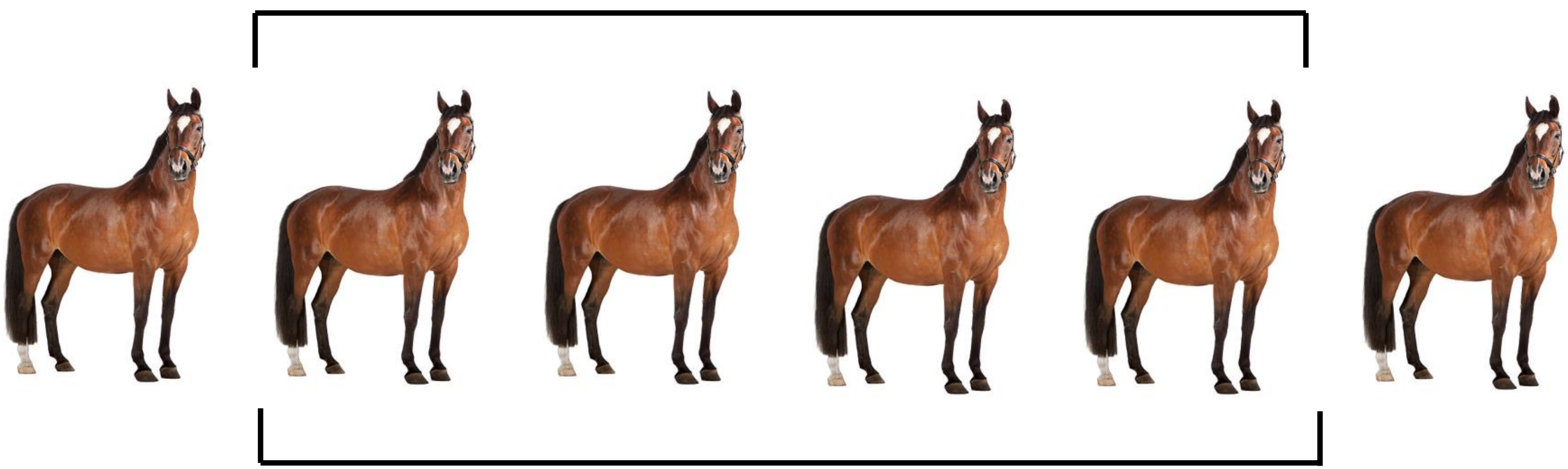


Inductive hypothesis: $n = k+1$

All Horses are the Same Color

Prove $P(k+1)$ = "All groups of $k+1$ horses always have the same color"

These horses in the middle were in both sets



And we said that both horses on the ends are the same color as these overlapping horses

Inductive hypothesis: $n = k+1$

All Horses are the Same Color

Prove $P(k+1)$ = “All groups of $k+1$ horses always have the same color”

So all $k+1$ horses have the same color!

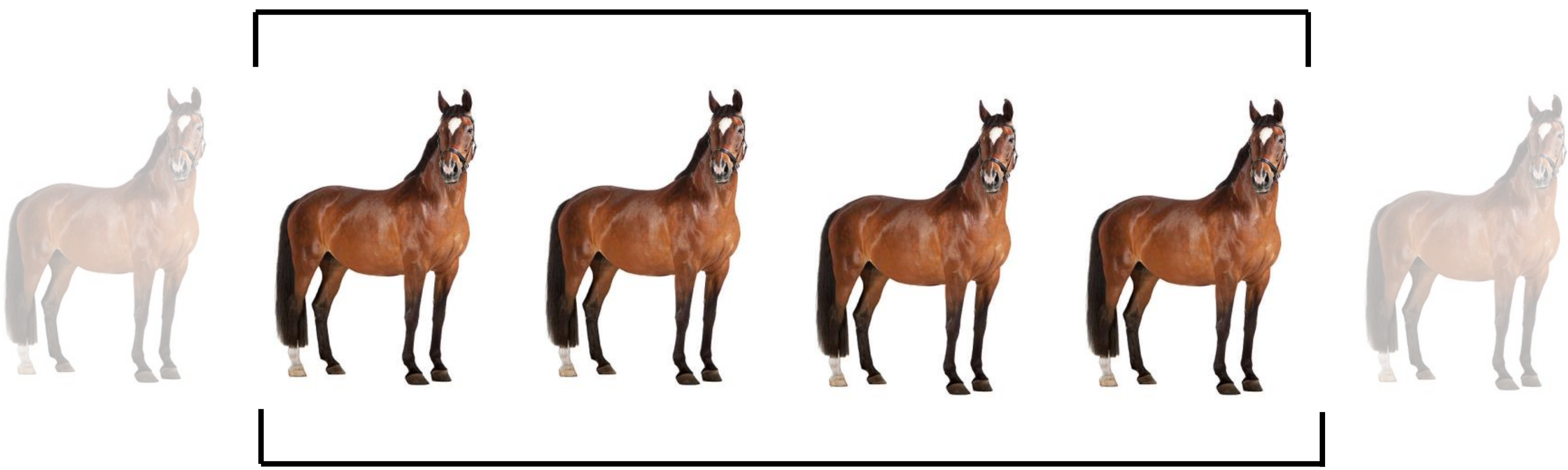
Inductive hypothesis: $n = k+1$

What's going on here?

All Horses are the Same Color

Prove $P(k+1)$ = "All groups of $k+1$ horses always have the same color"

These horses in the middle were in both sets



Inductive hypothesis: $n = k+1$

All Horses are the Same Color

Prove $P(k+1)$ = “All groups of $k+1$ horses always have the same color”

These horses in the middle were in both sets

But what if there are no such horses?

Inductive hypothesis: $n = k+1$

All Horses are the Same Color

$P(n)$ = "All groups of n horses always have the same color"

$$P(1) \rightarrow P(2)$$

All Horses are the Same Color

$P(n)$ = "All groups of n horses always have the same color"

By $P(1)$, this 1 horse has the same color

$$P(1) \rightarrow P(2)$$

All Horses are the Same Color

$P(n)$ = "All groups of n horses always have the same color"

By $P(1)$, this 1 horse has the same color

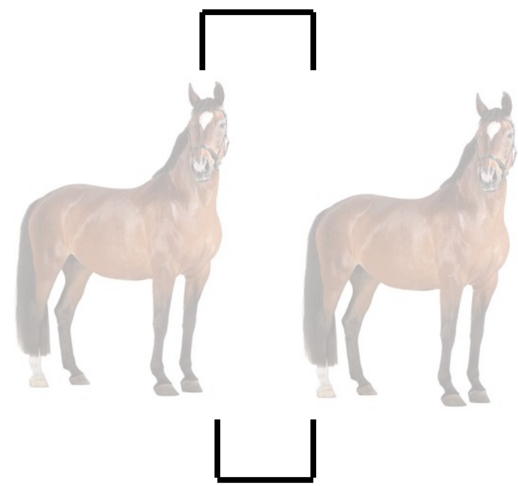
By $P(1)$, this 1 horse has the same color

$$P(1) \rightarrow P(2)$$

All Horses are the Same Color

$P(n)$ = "All groups of n horses always have the same color"

These horses in the middle (??) were in both sets



$$P(1) \rightarrow P(2)$$

Theorem: All horses are the same color.

Proof: Let $P(n)$ be the statement "all groups of n horses are the same color." We will prove by induction that $P(n)$ holds for all natural numbers n, from which the theorem follows.

As our base case, we prove $P(0)$, that all groups of 0 horses are the same color. This statement is vacuously true because there are no horses.

For the inductive step, assume that for an arbitrary natural number k that $P(k)$ is true and that all groups of k horses are the same color. Now consider a group of $k+1$ horses. Exclude the last horse and look only at the first k horses. By the inductive hypothesis, these horses are the same color. Next, exclude the first horse and look only at the last k horses. Again we see by the inductive hypothesis that these horses are the same color.

Therefore, the first horse is the same color as the non-excluded horses, who in turn are the same color as the last horse. **Hence the first horse excluded, the non-excluded horses, and last horse excluded are all of the same color**. Thus $P(k+1)$ holds, completing the induction. ■

Complete Induction

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

$P(0)$ is true

If it starts true…

and

…and it stays true…

for any $k \in \mathbb{N}$, if $P(0)$, $P(1)$, …, and $P(k)$ are true, then $P(k+1)$ is true

then

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

…then it's always true.

Mathematical Induction

You can write proofs using the principle of mathematical induction as follows:

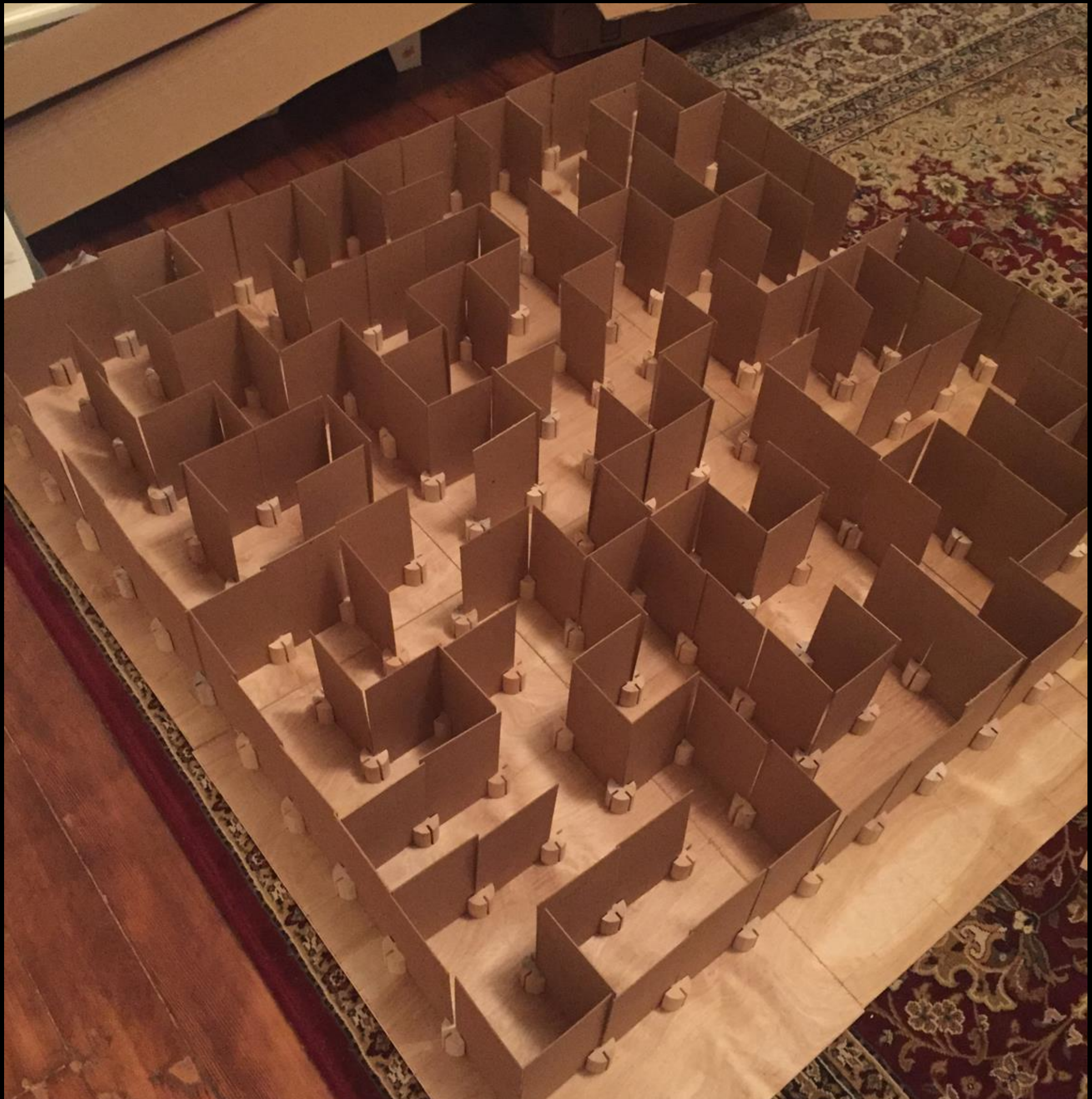
- Define some predicate $P(n)$ to prove by induction on n.
- Choose and prove a base case (probably, but not always, $P(0)$).
- Pick an arbitrary $k \in \mathbb{N}$ and assume that $P(k)$ is true.
- Prove $P(k+1)$.
- Conclude that $P(n)$ holds for all $n \in \mathbb{N}$.

Complete Induction

You can write proofs using the principle of ***complete*** induction as follows:

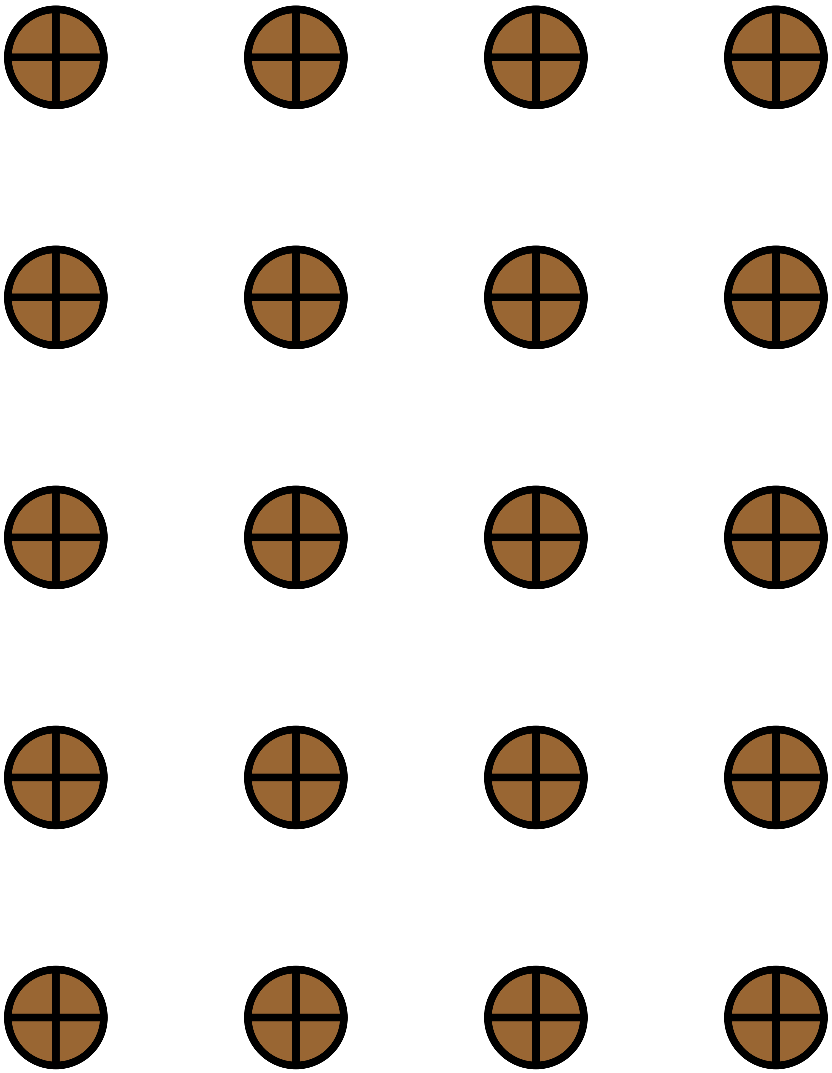
- Define some predicate $P(n)$ to prove by induction on n.
- Choose and prove a base case (probably, but not always, $P(0)$).
- Pick an arbitrary $k \in \mathbb{N}$ and assume that $\boldsymbol{P(0), P(1), P(2), \ldots,}$ **and** $\boldsymbol{P(k)}$ are all true.
- Prove $P(k+1)$.
- Conclude that $P(n)$ holds for all $n \in \mathbb{N}$.

A Motivating Example: ***Rat Mazes***



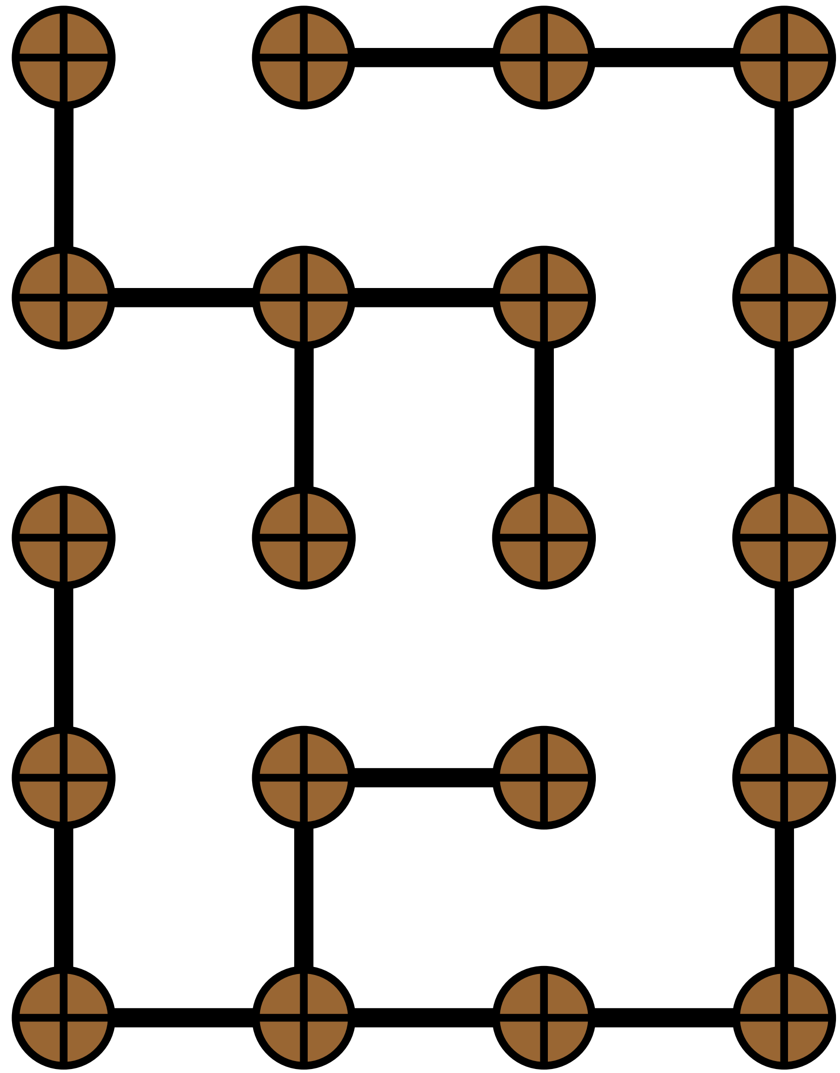
Rat Mazes

Suppose you want to make a rat maze consisting of an $n \times m$ grid of pegs with slats between them.



Rat Mazes

Suppose you want to make a rat maze consisting of an $n \times m$ grid of pegs with slats between them.



Rat Mazes

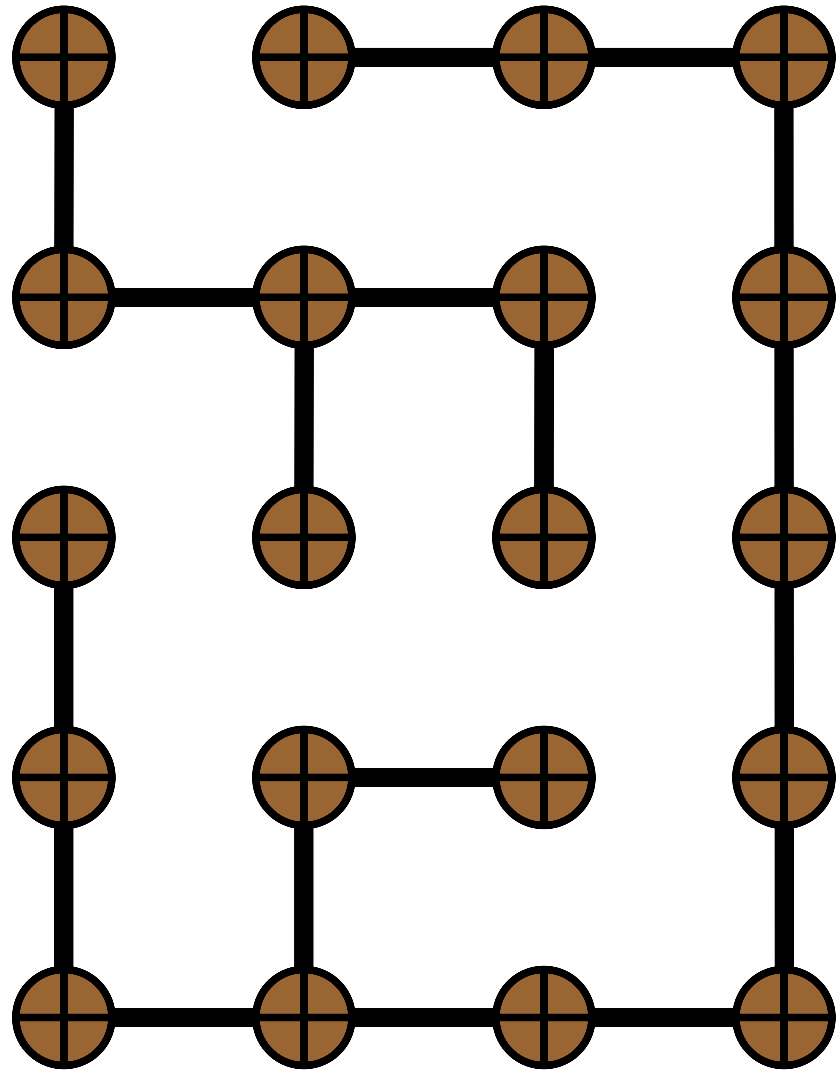
Suppose you want to make a rat maze consisting of an $n \times m$ grid of pegs with slats between them.

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.

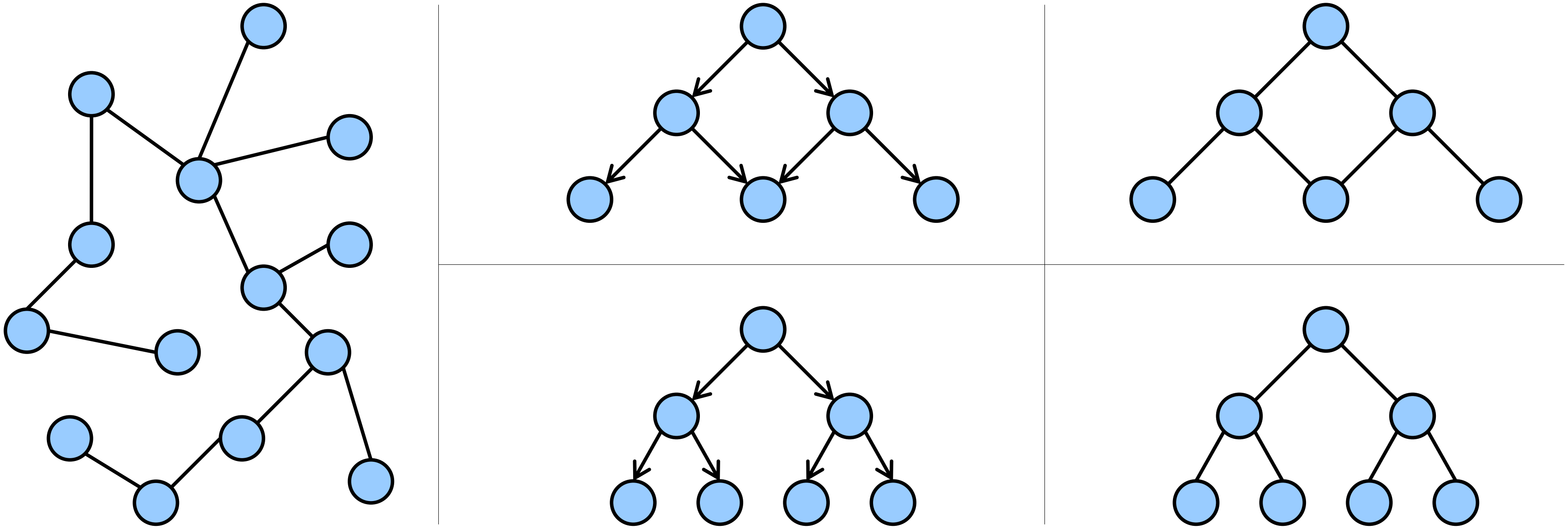


Question: If you have an $n \times m$ grid of pegs, how many slats do you need to make?

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

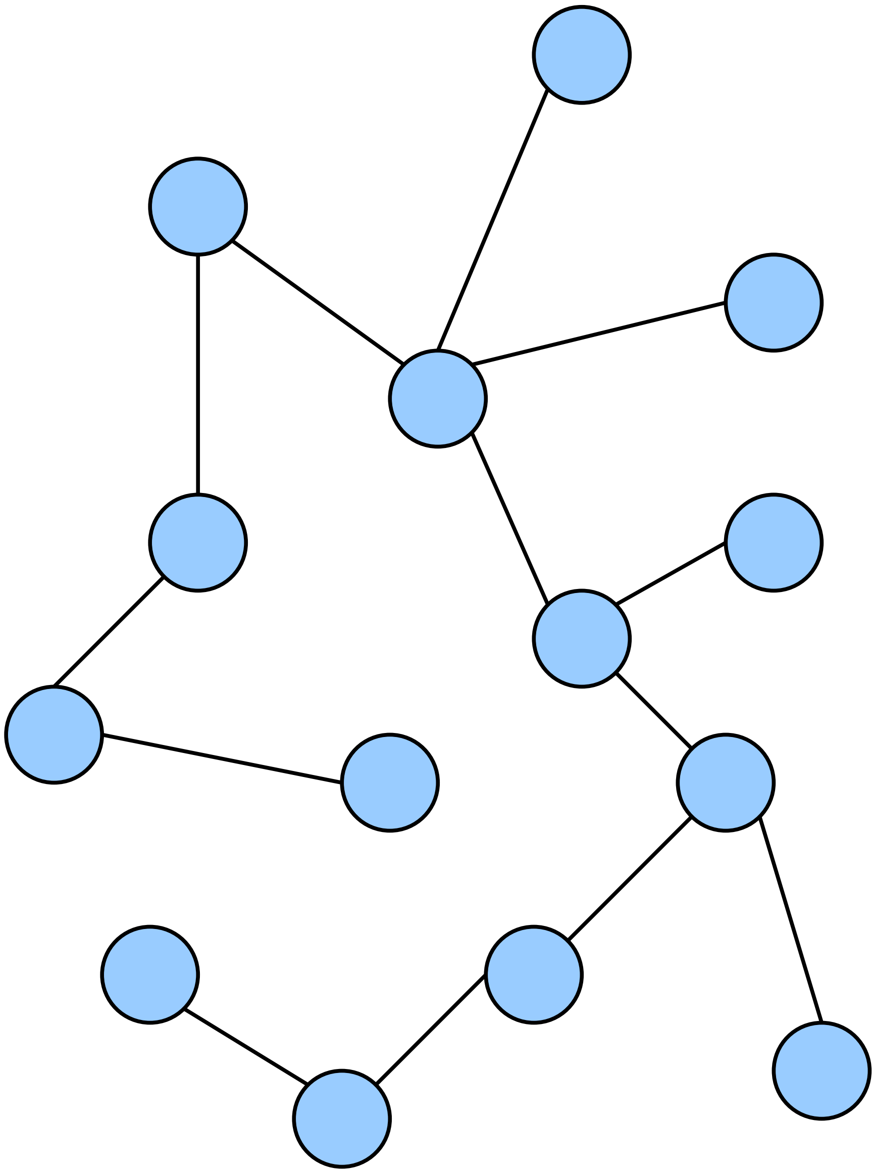
A ***tree*** is a connected, nonempty graph with no simple cycles.

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



Trees

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

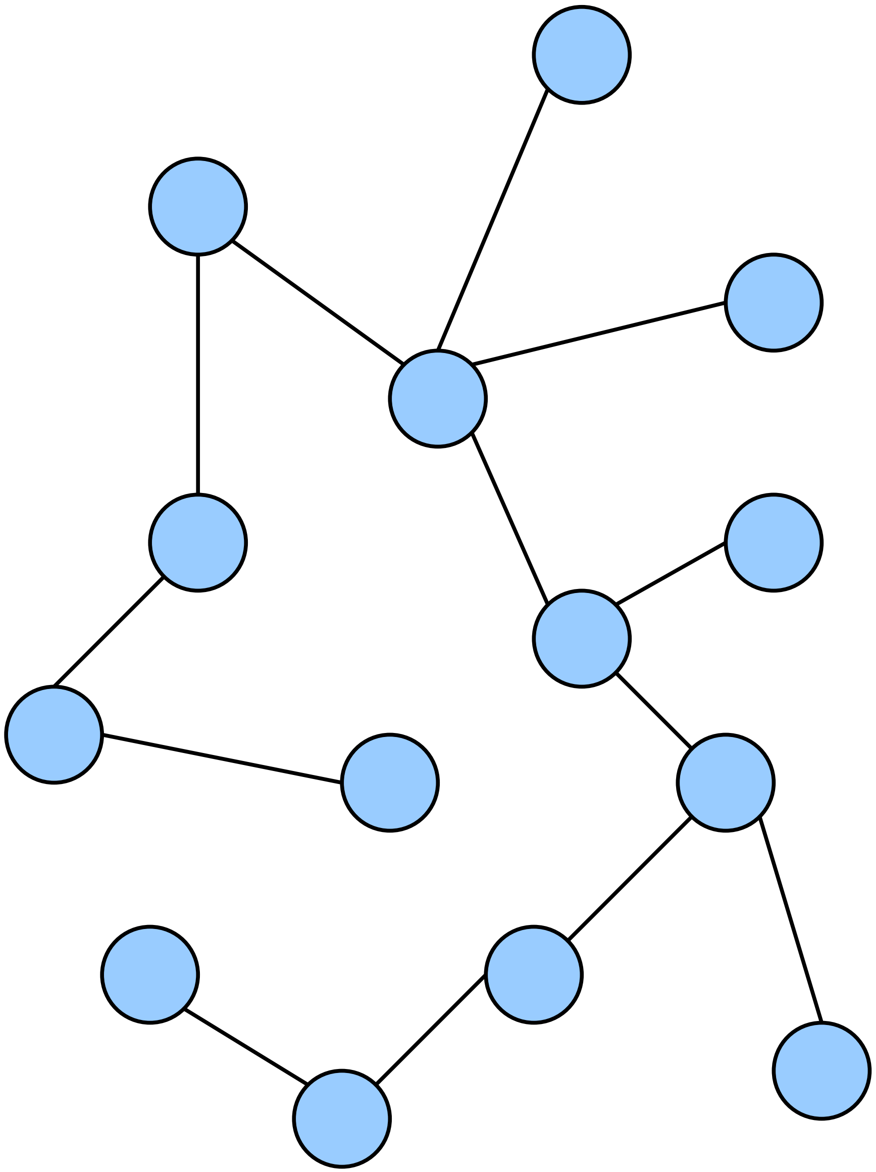


Trees

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)

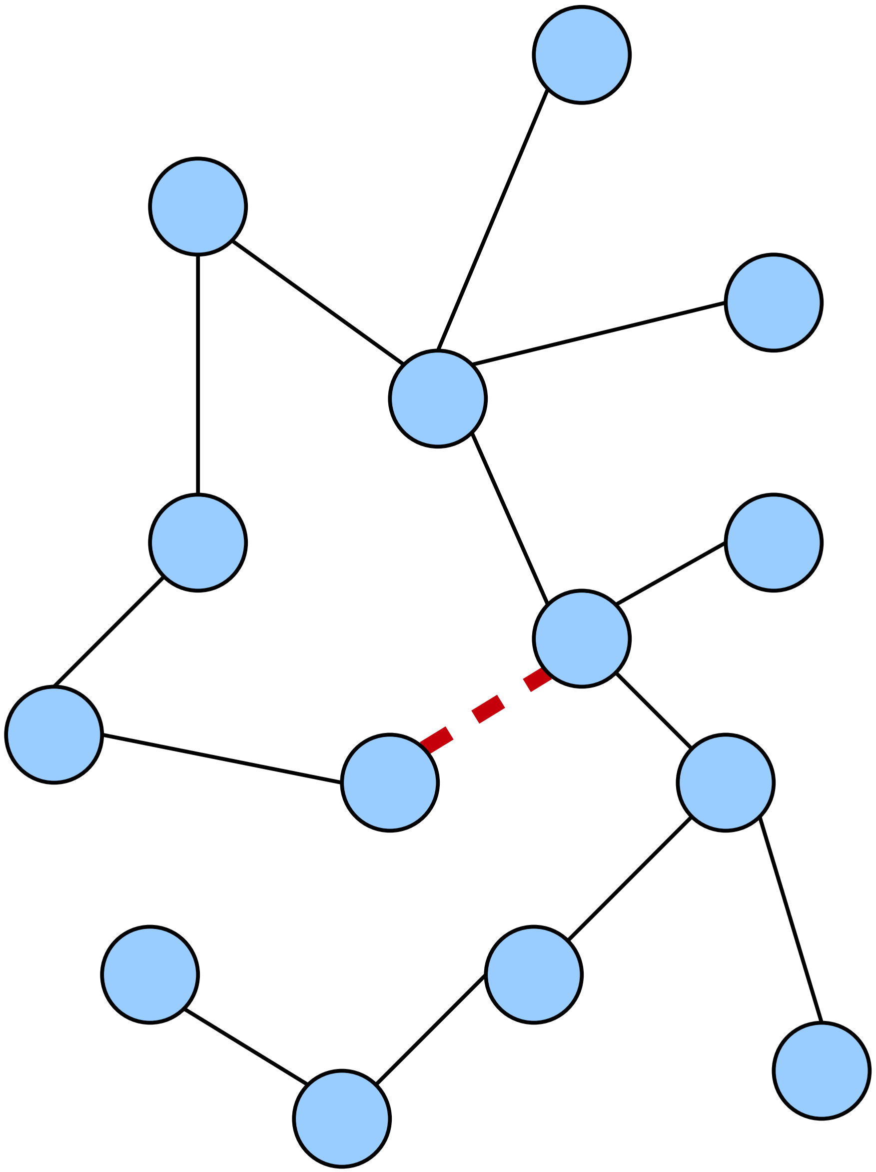


Trees

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)

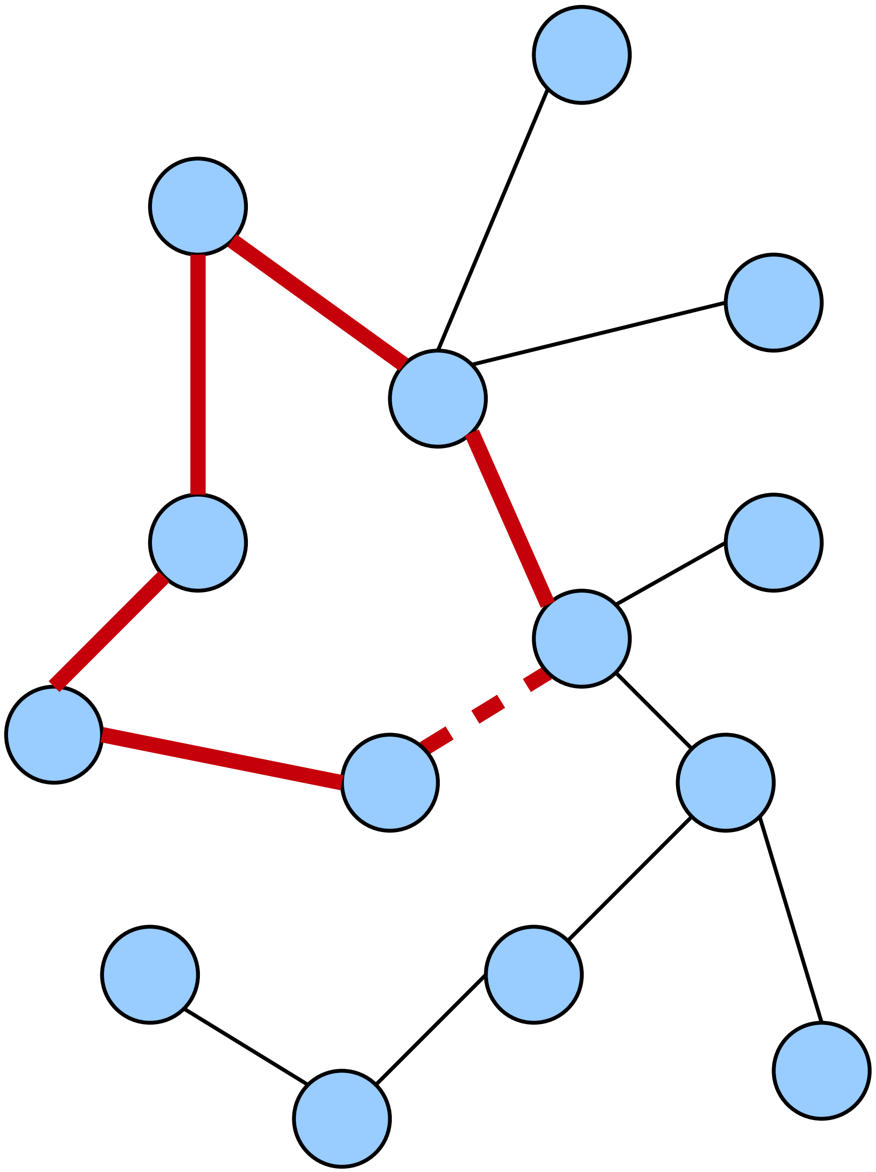


Trees

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)



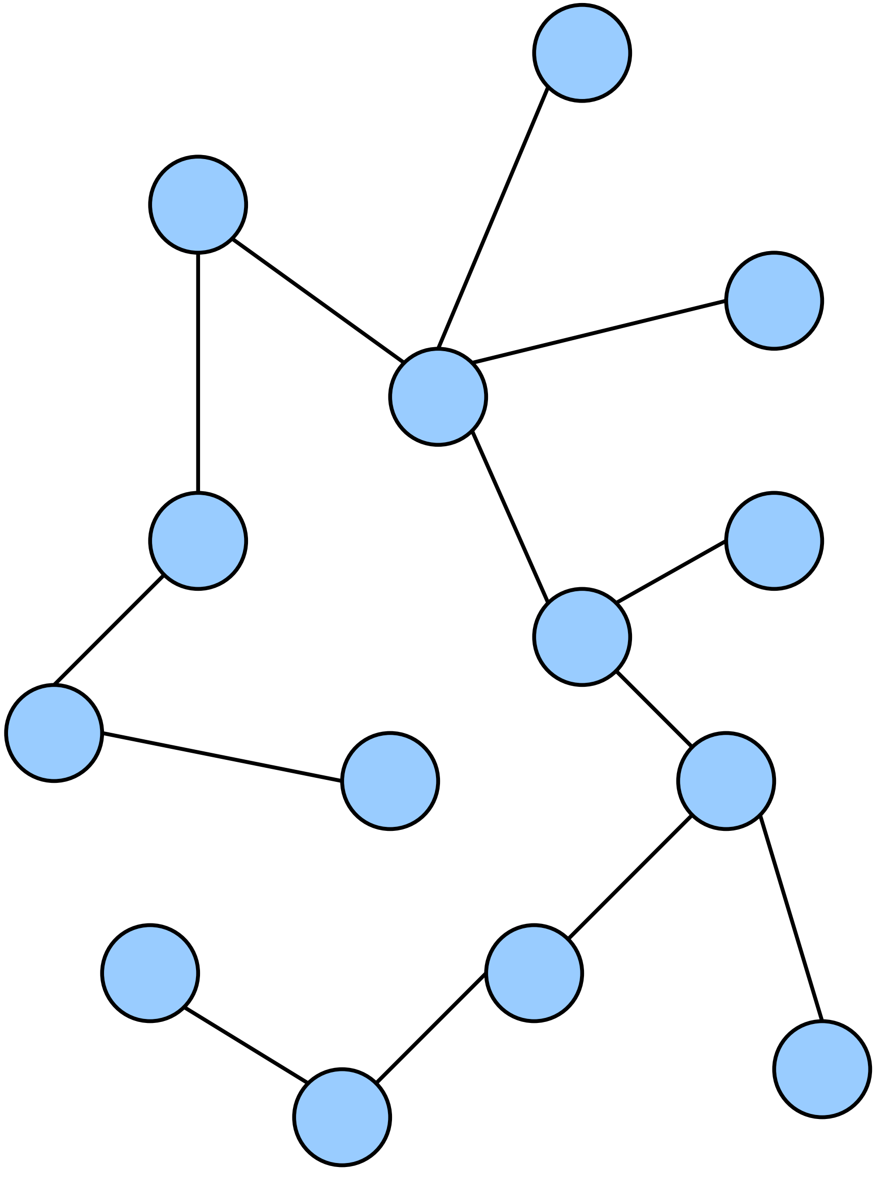
Trees

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)



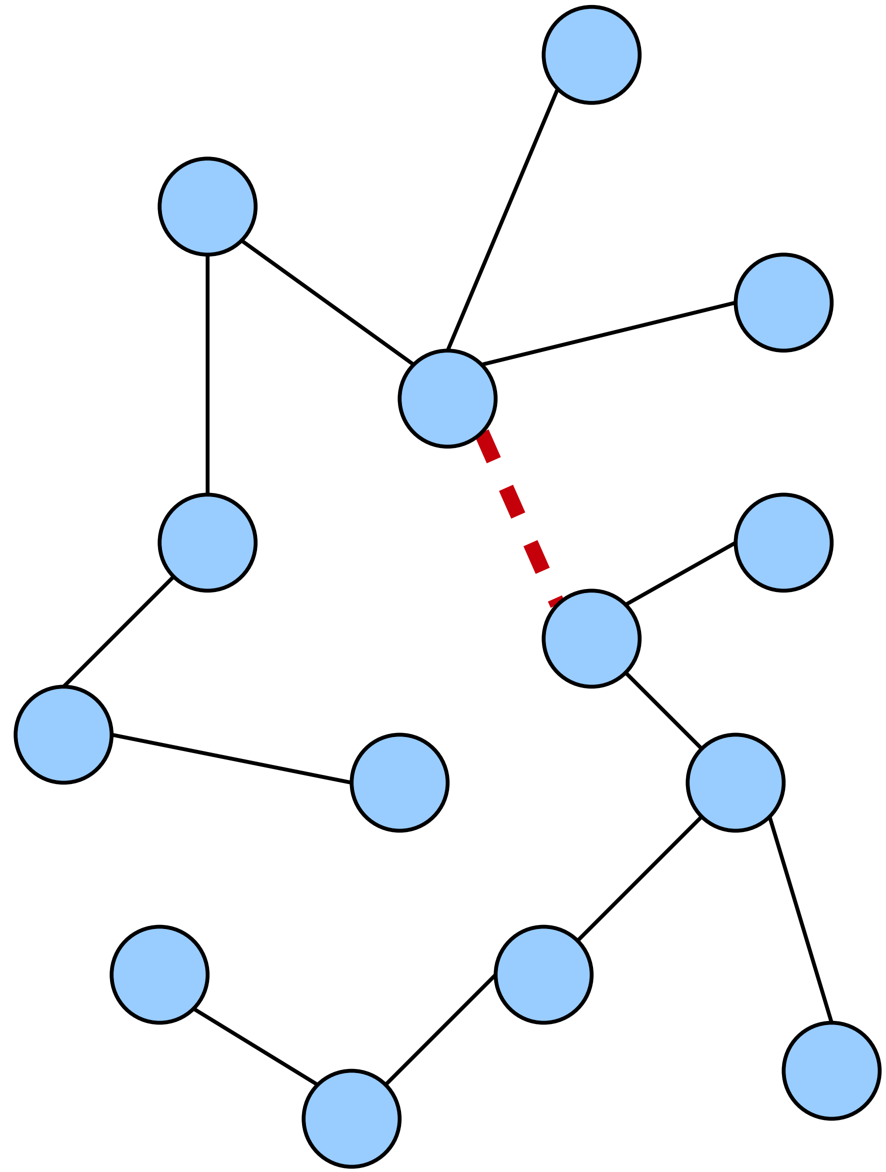
Trees

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)



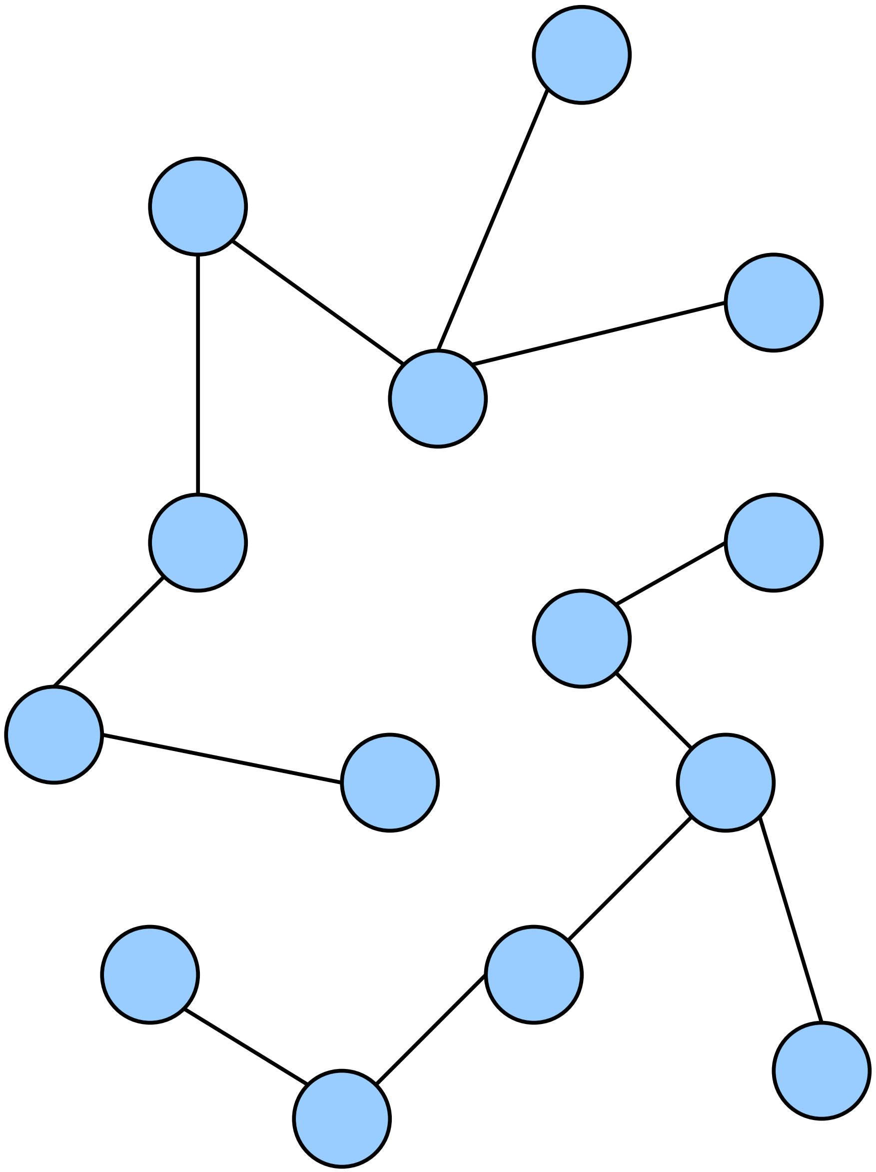
Trees

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)



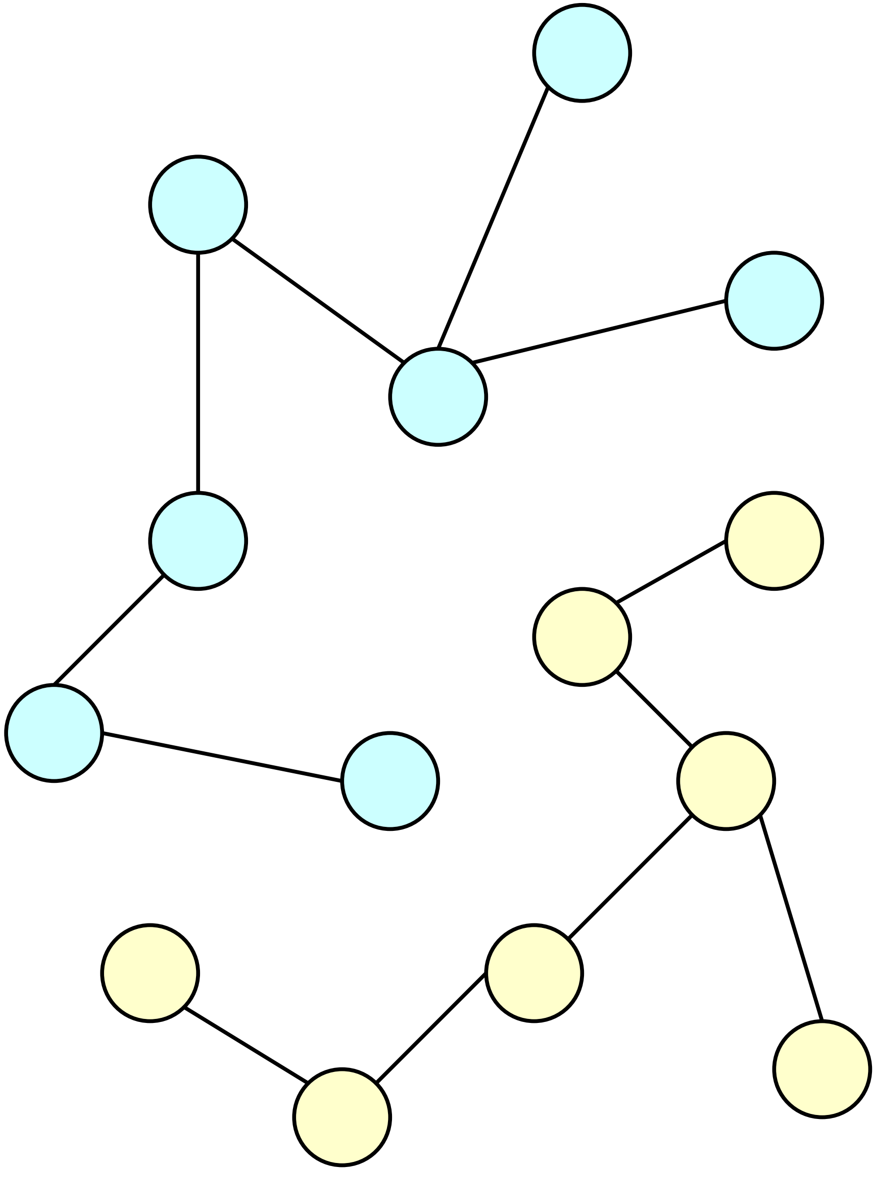
Trees

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)



Trees

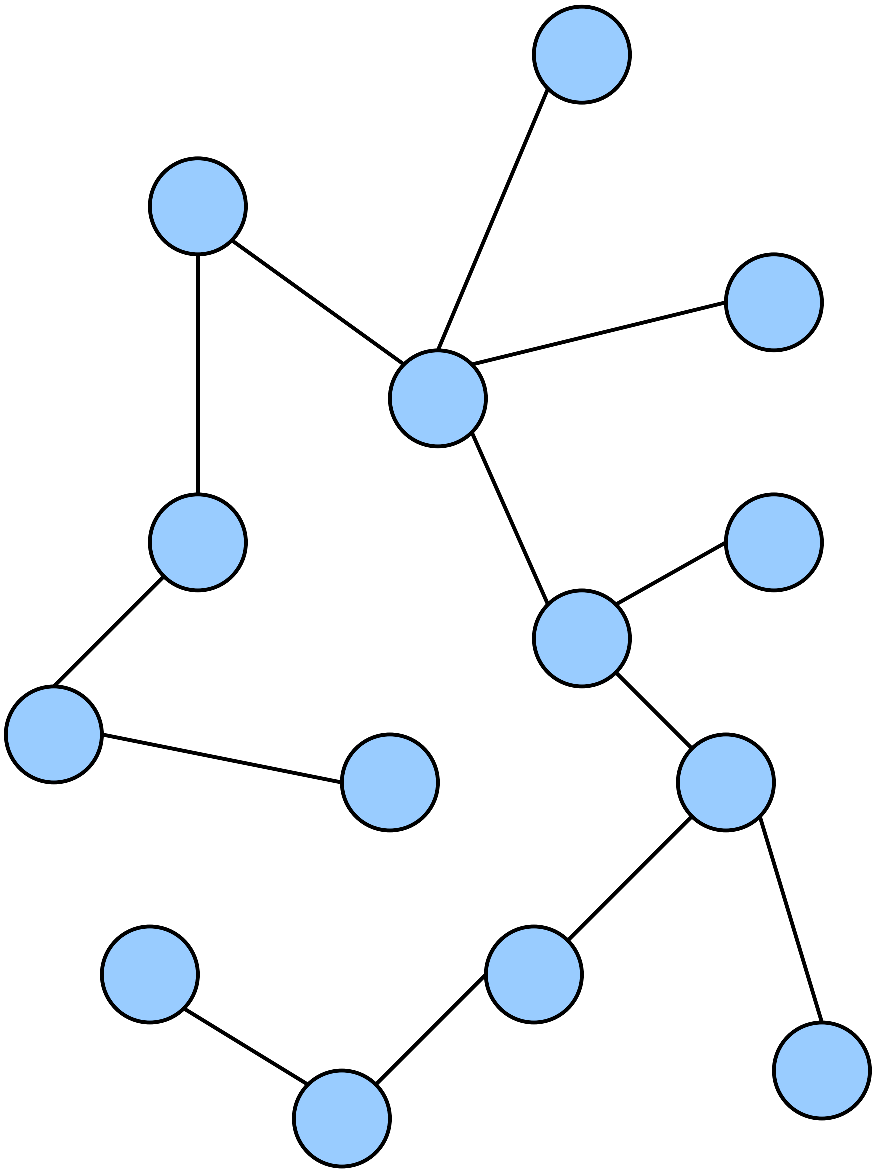
A ***tree*** is a connected, nonempty graph with no simple cycles.

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)

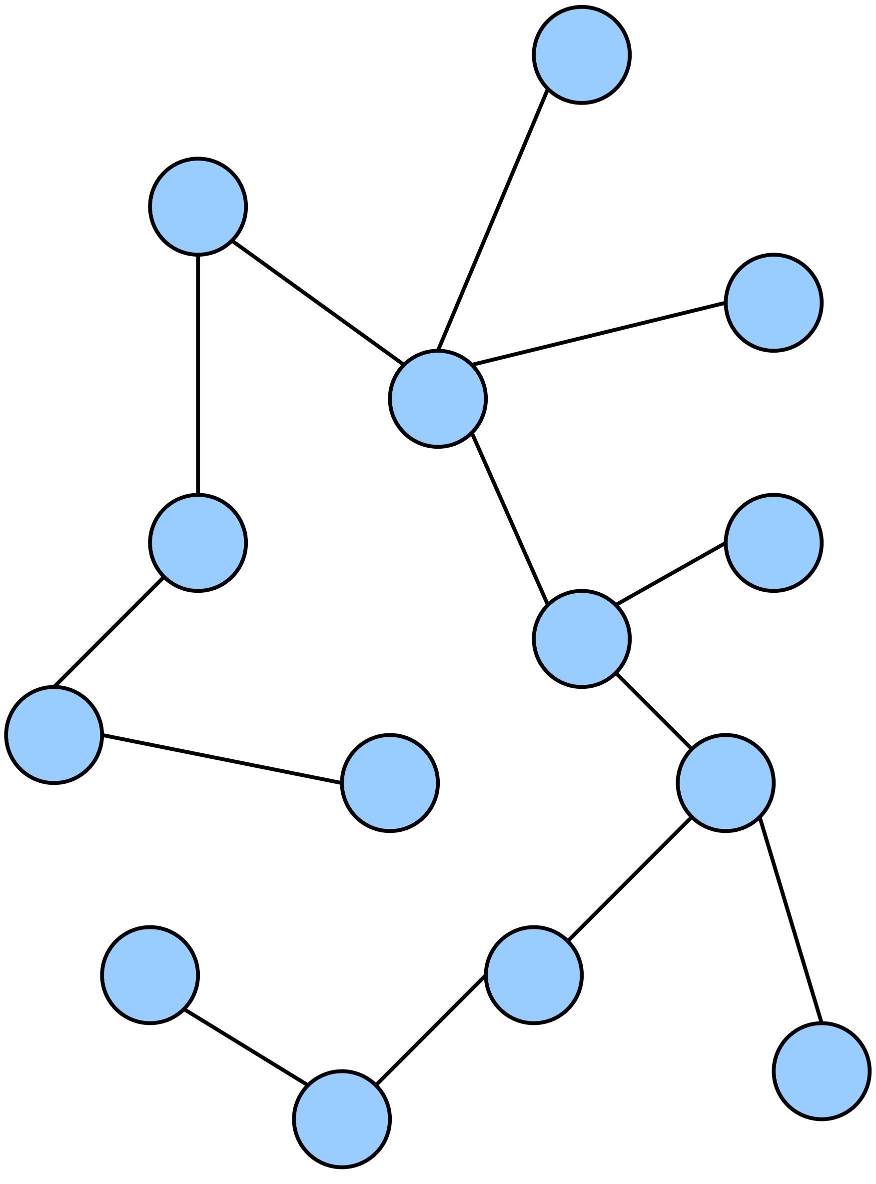
Proofs of these results are in the course reader if you're interested. They're also great exercises.



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.

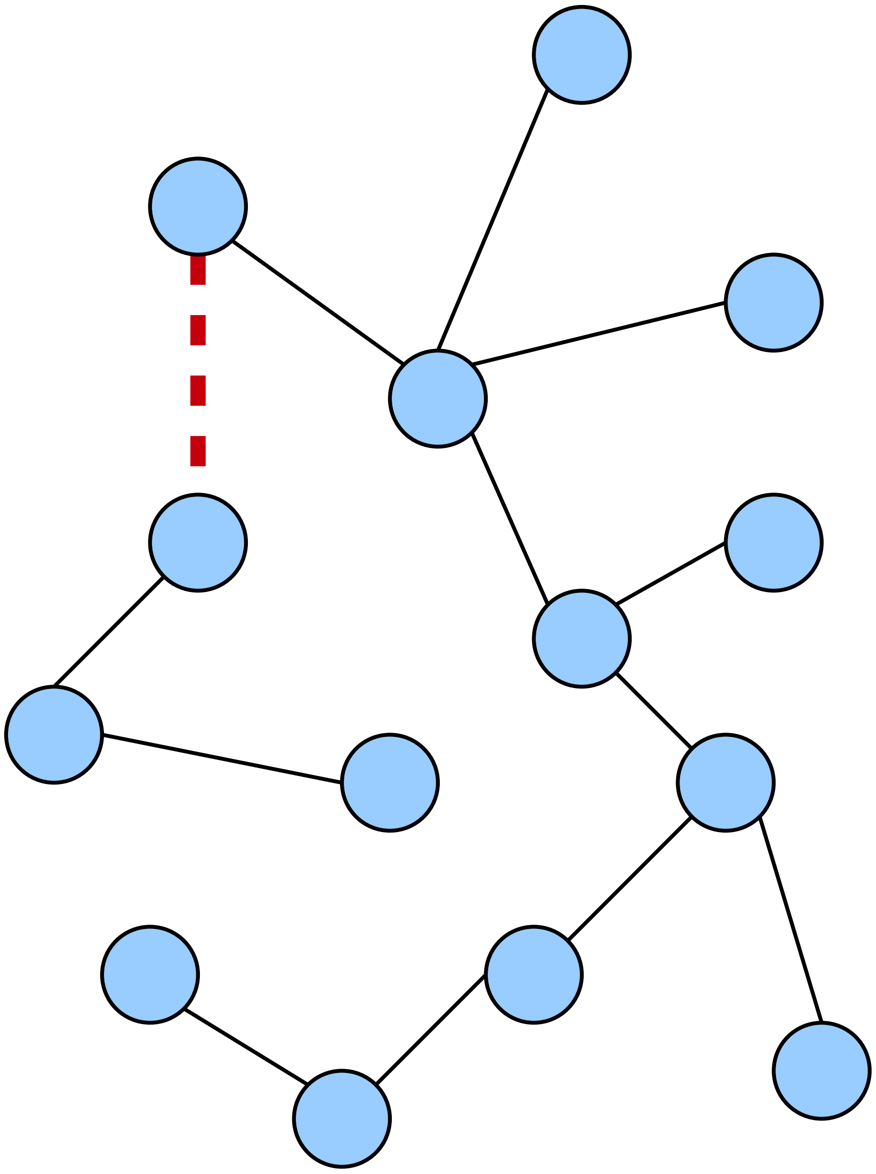
Proof: Left as an exercise to the reader. ☺



Trees

Theorem: If T is a tree with at least two nodes, then deleting any edge from T splits T into two nonempty trees T_1 and T_2.

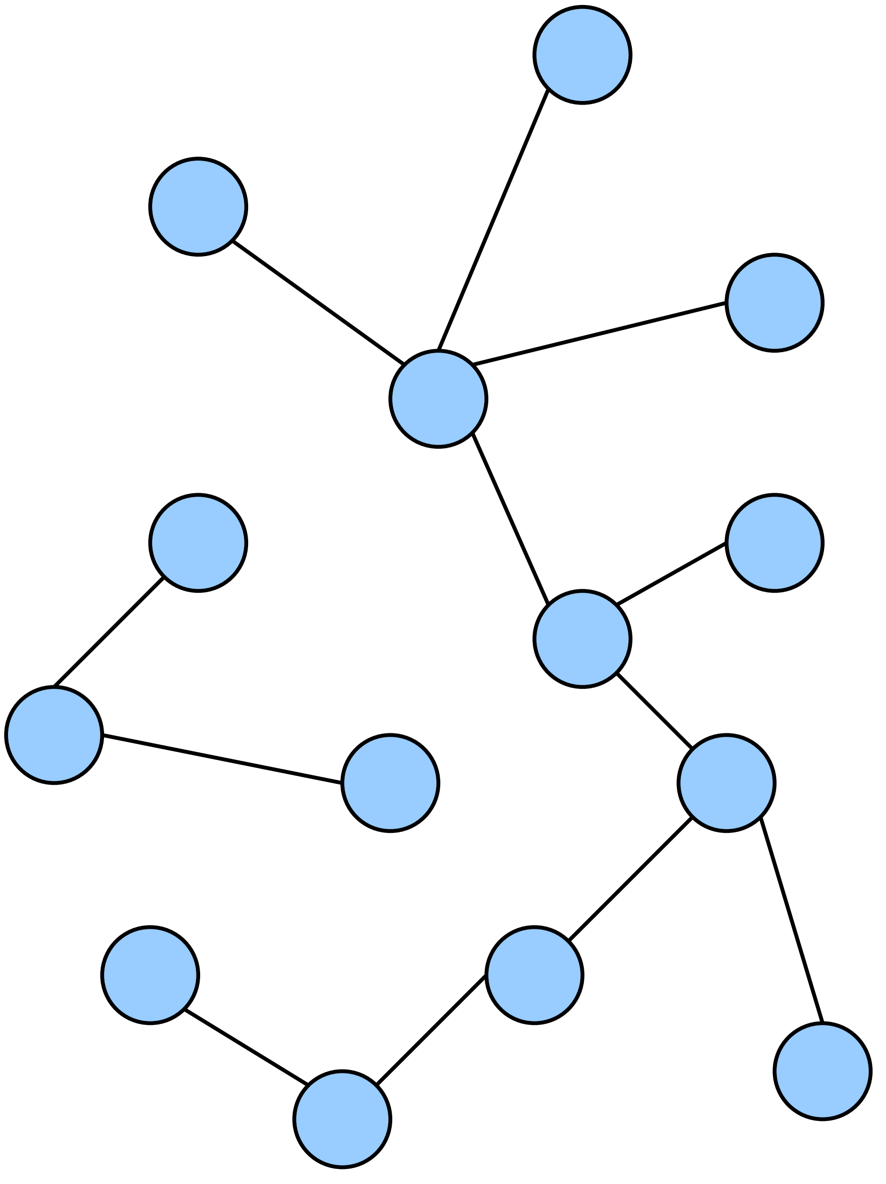
Proof: Left as an exercise to the reader. ☺



Trees

Theorem: If T is a tree with at least two nodes, then deleting any edge from T splits T into two nonempty trees T_1 and T_2.

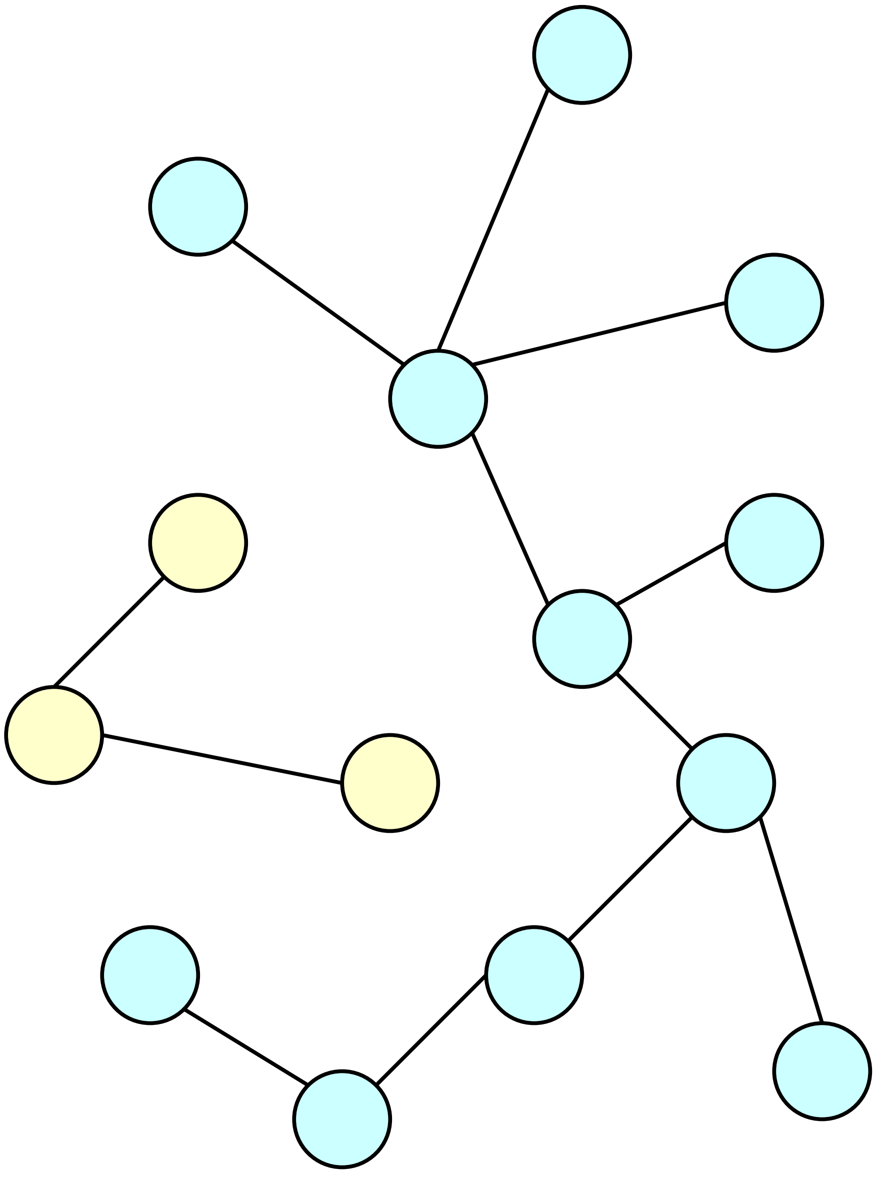
Proof: Left as an exercise to the reader. ☺



Trees

Theorem: If T is a tree with at least two nodes, then deleting any edge from T splits T into two nonempty trees T_1 and T_2.

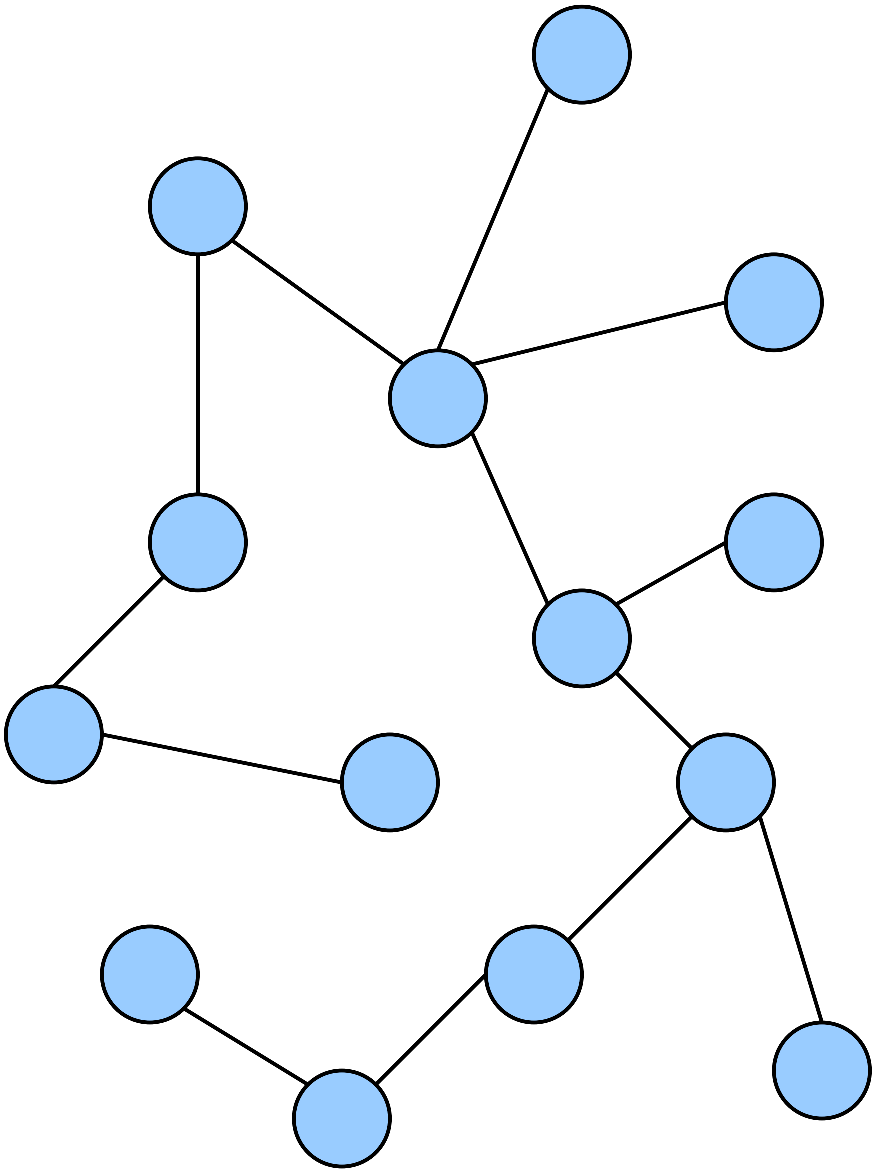
Proof: Left as an exercise to the reader. ☺



Trees

Theorem: If T is a tree with at least two nodes, then deleting any edge from T splits T into two nonempty trees T_1 and T_2.

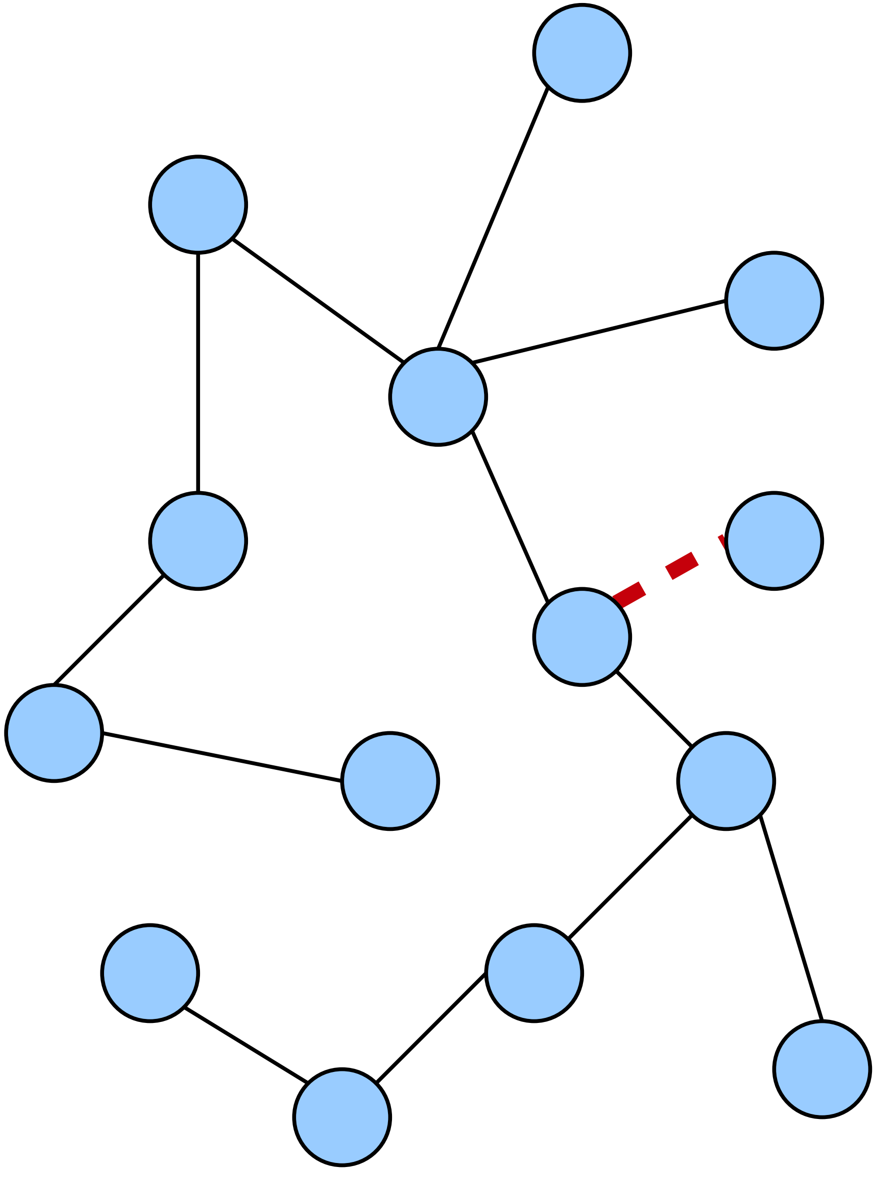
Proof: Left as an exercise to the reader. ☺



Trees

Theorem: If T is a tree with at least two nodes, then deleting any edge from T splits T into two nonempty trees T_1 and T_2.

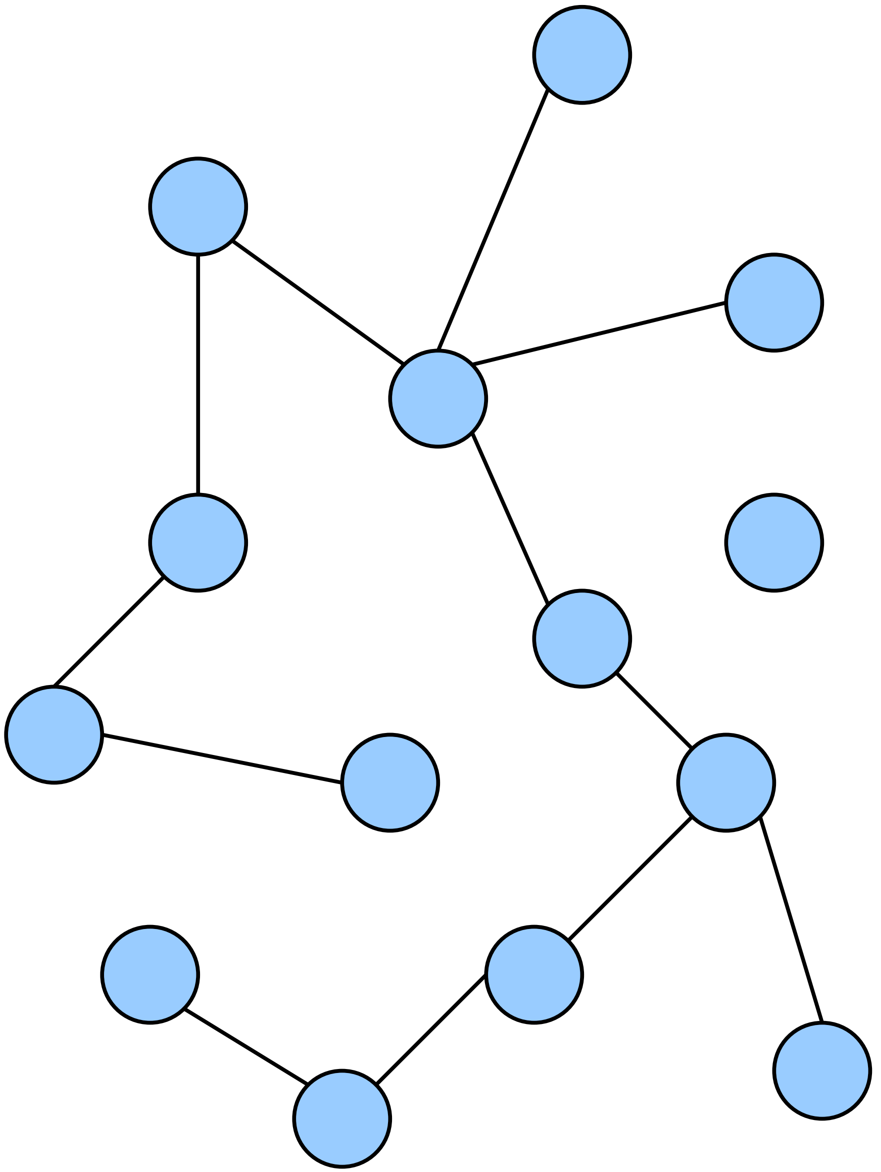
Proof: Left as an exercise to the reader. ☺



Trees

Theorem: If T is a tree with at least two nodes, then deleting any edge from T splits T into two nonempty trees T_1 and T_2.

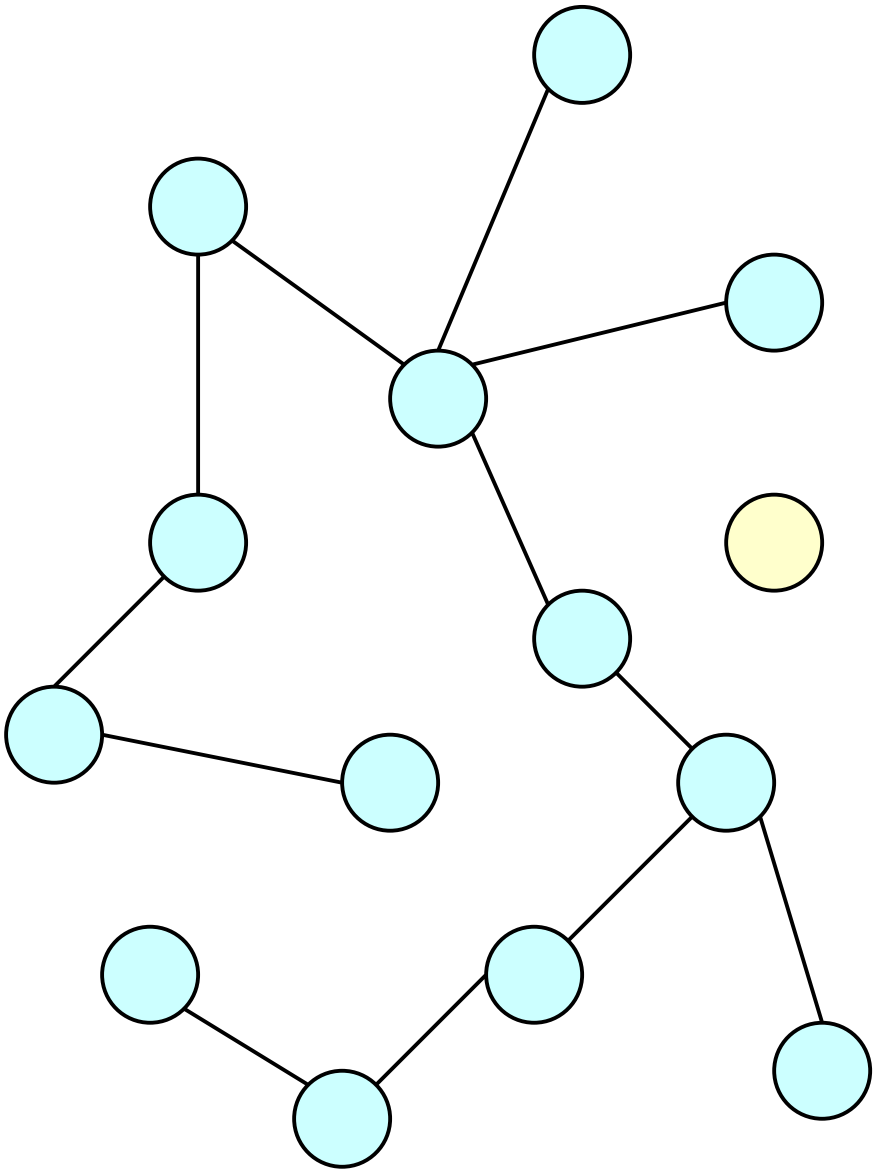
Proof: Left as an exercise to the reader. ☺



Trees

Theorem: If T is a tree with at least two nodes, then deleting any edge from T splits T into two nonempty trees T_1 and T_2.

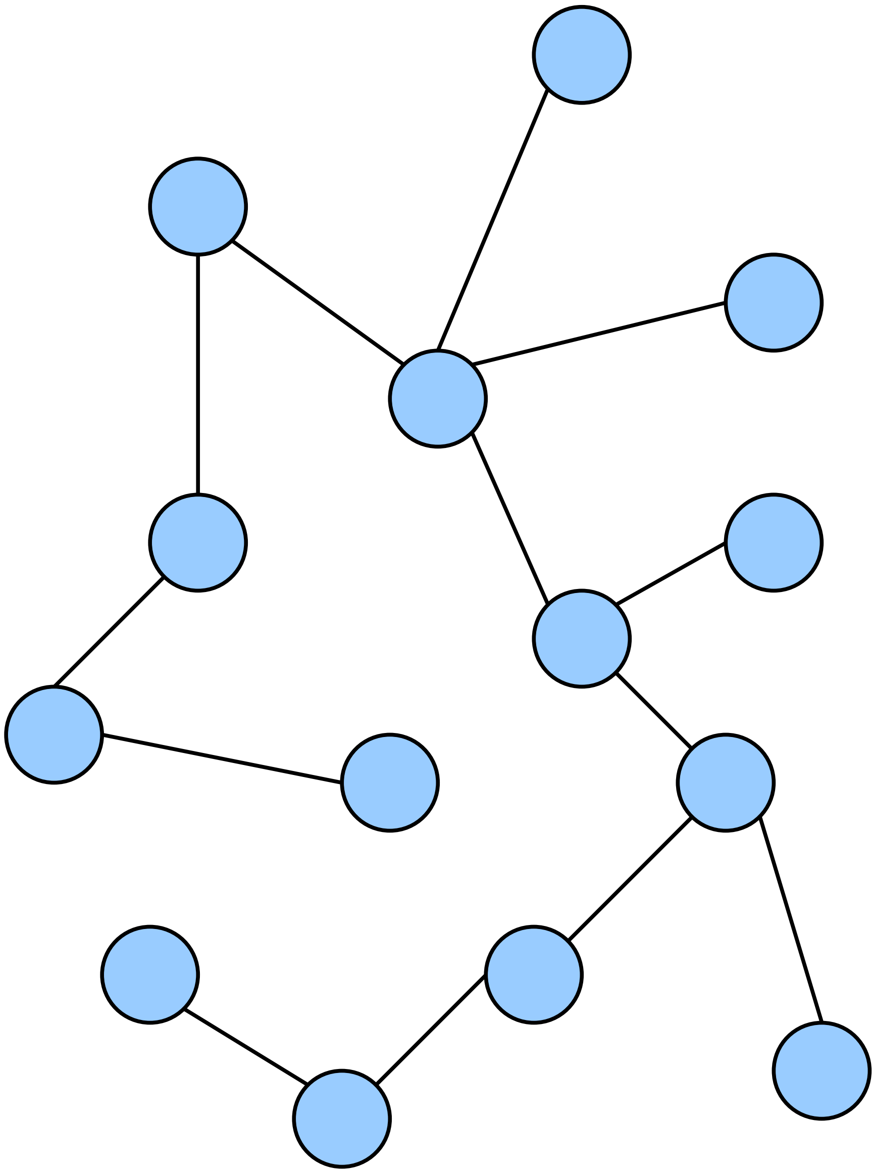
Proof: Left as an exercise to the reader. ☺



Trees

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

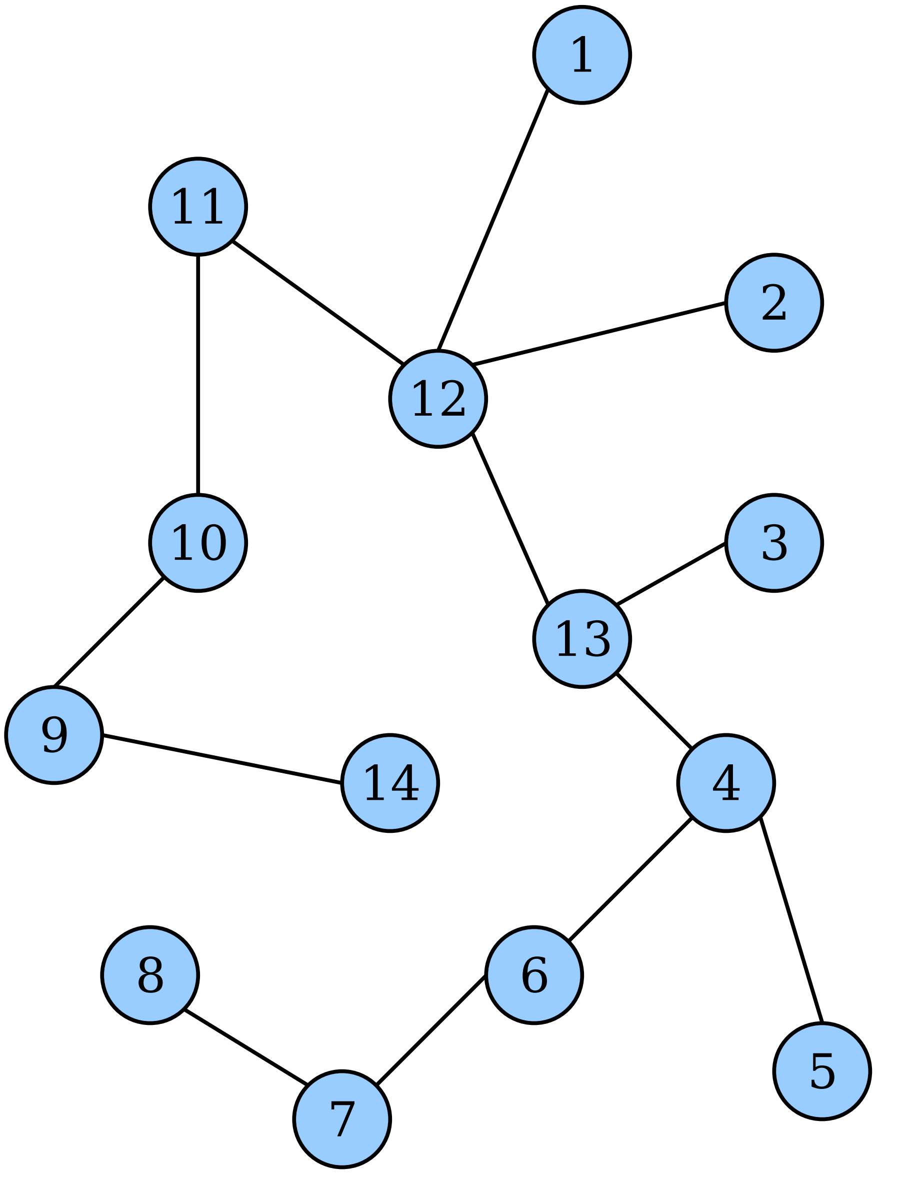
Proof: Up next!



Trees

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

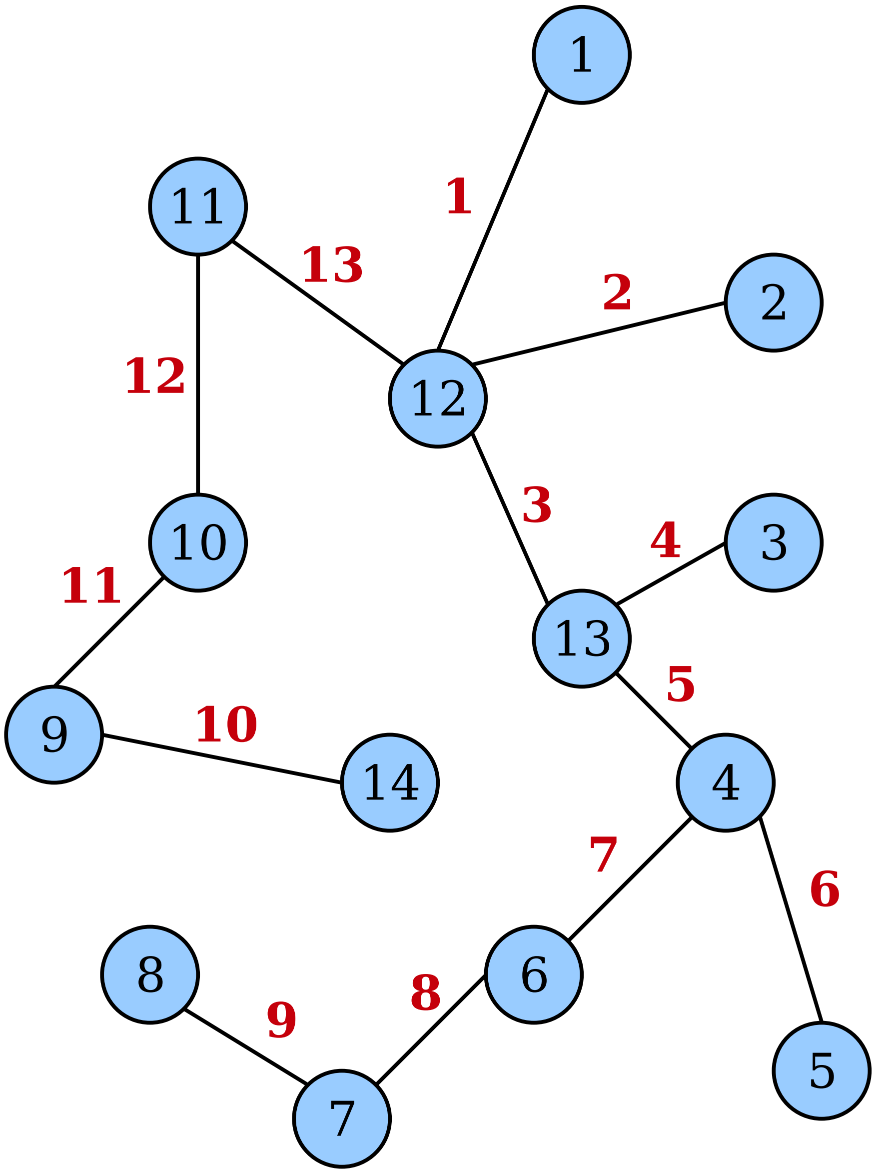
Proof: Up next!



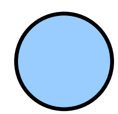
Trees

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

Proof: Up next!



Our Base Case



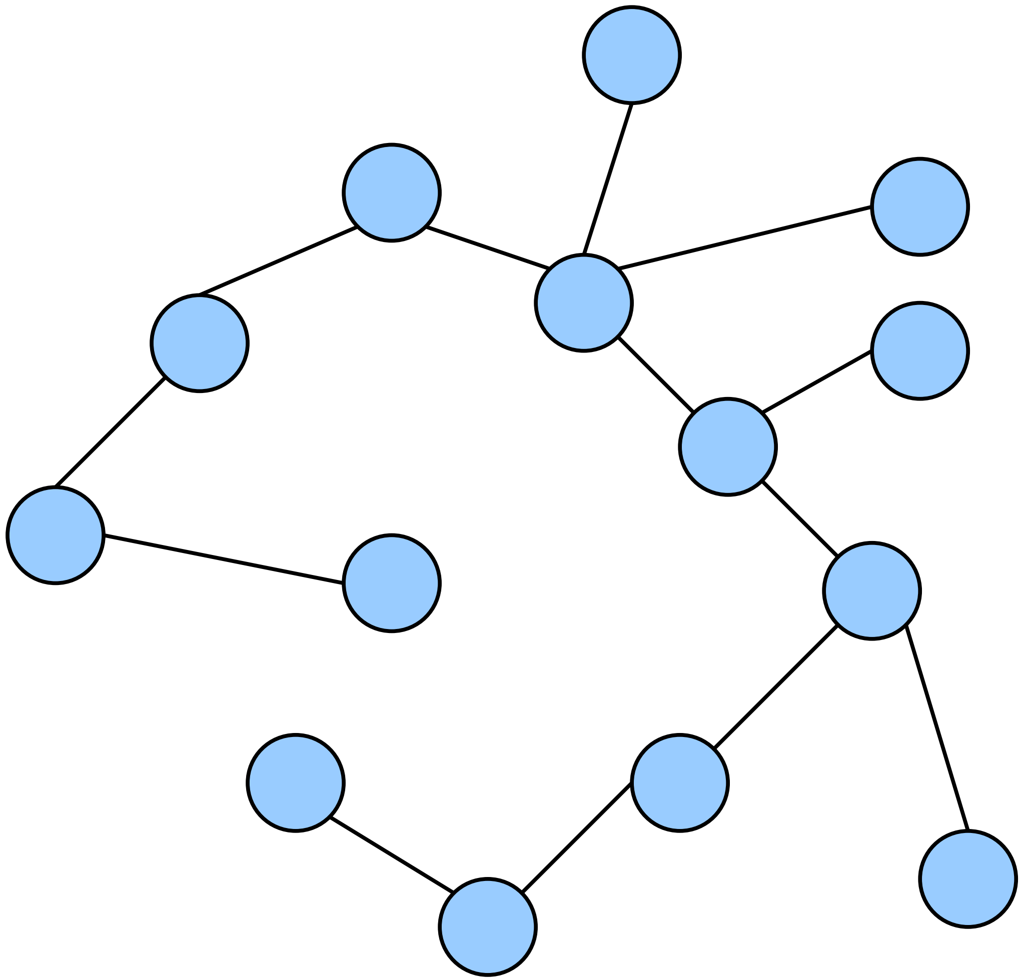
Assume any tree with at most *k* nodes has one more node than edge.

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

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

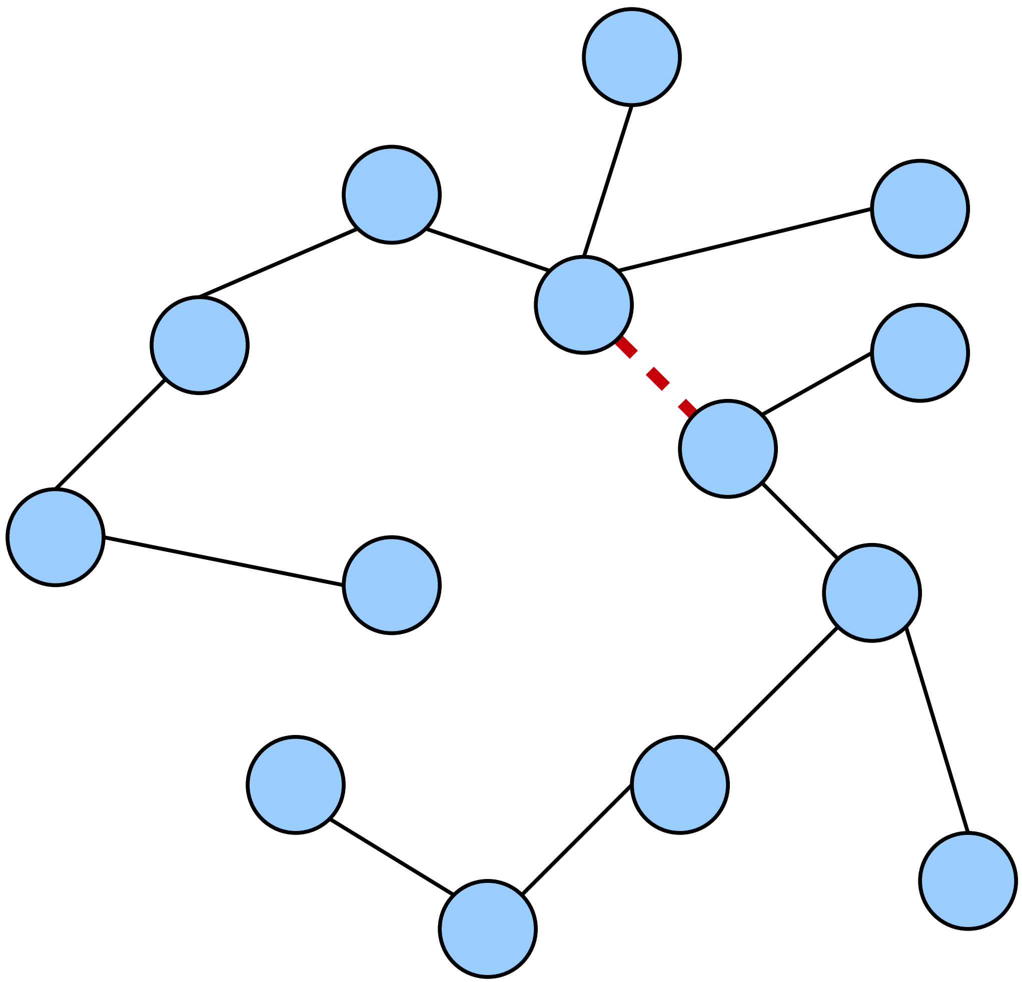
Assume any tree with at most *k* nodes has one more node than edge.

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



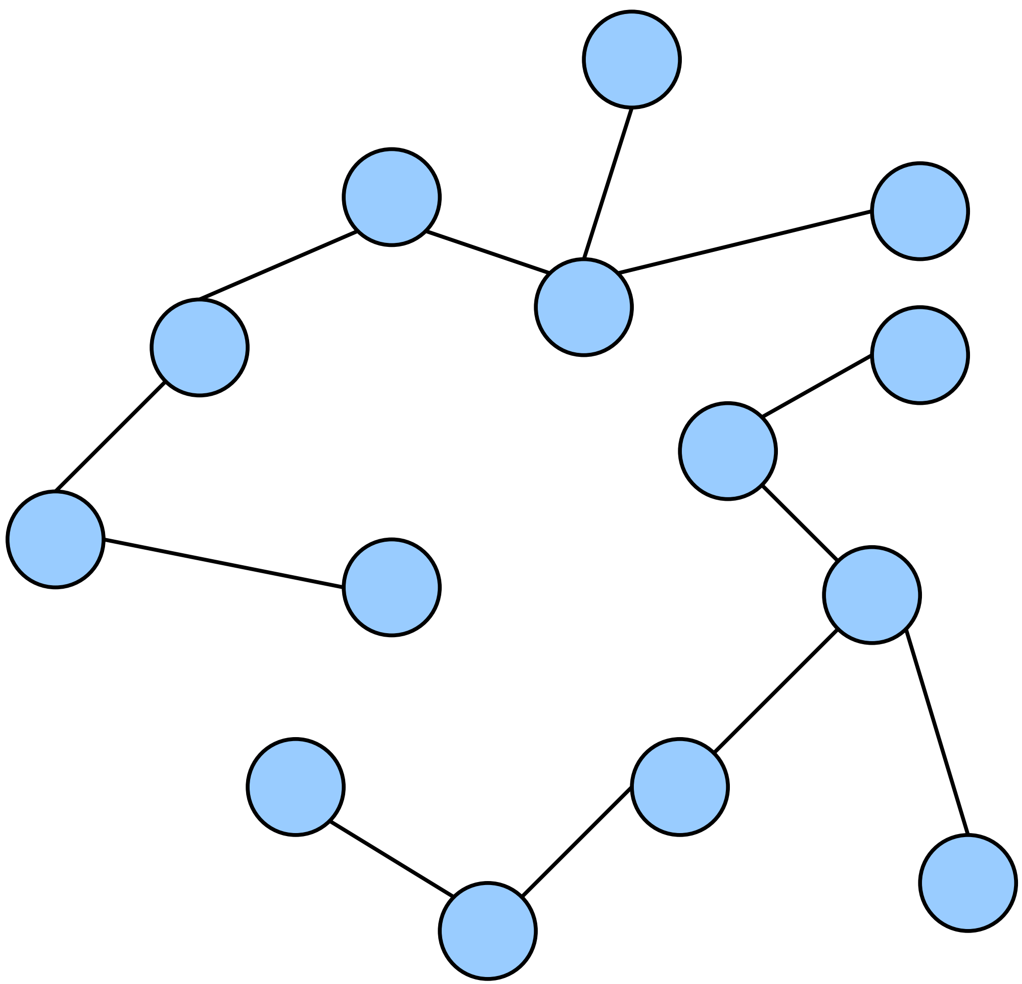
Assume any tree with at most *k* nodes has one more node than edge.

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



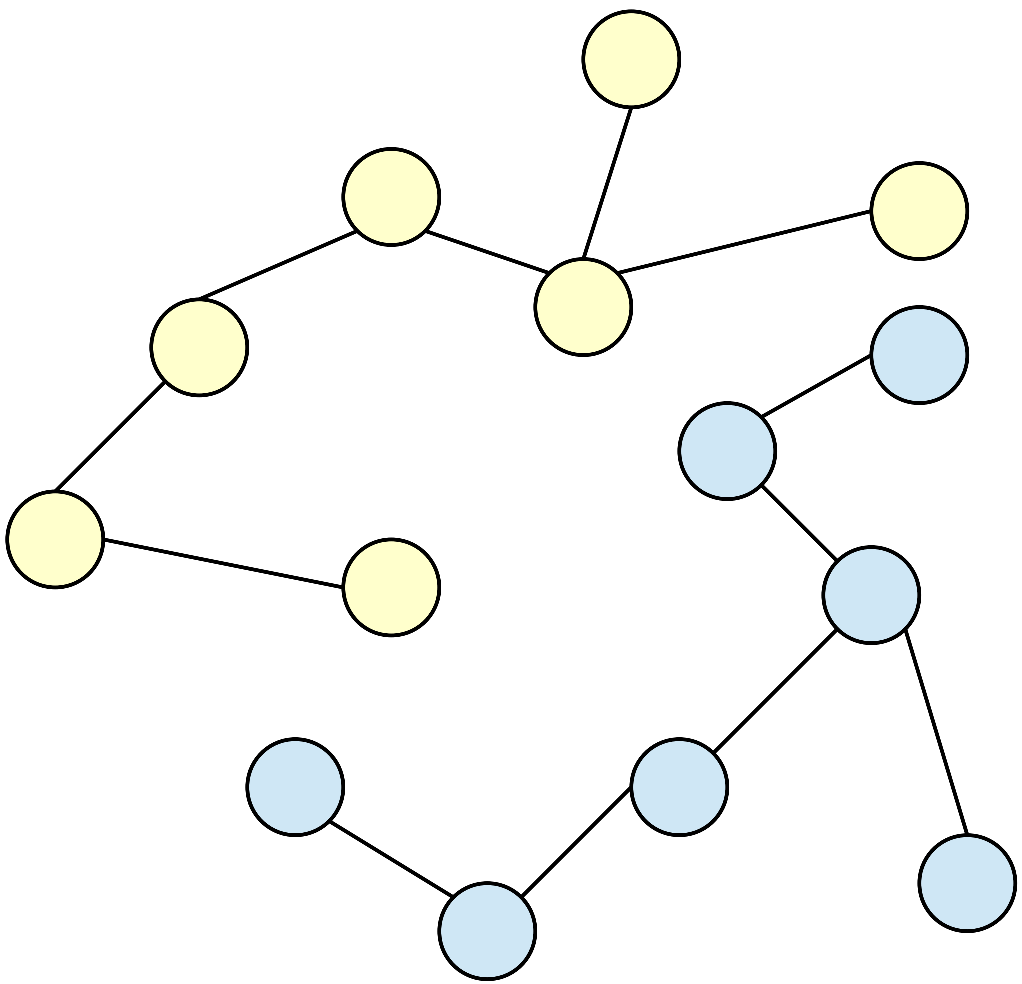
Assume any tree with at most *k* nodes has one more node than edge.

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



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

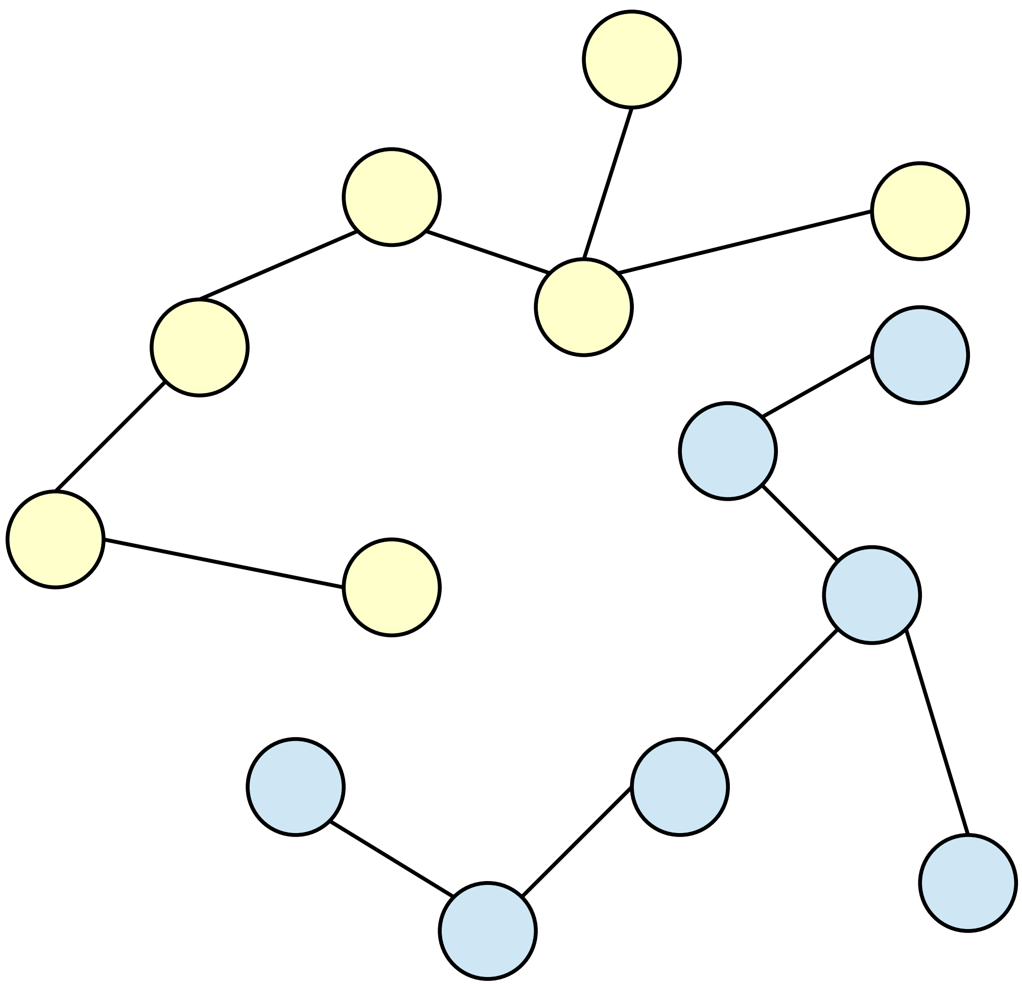
Consider an arbitrary tree with *k*+1 nodes.



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.

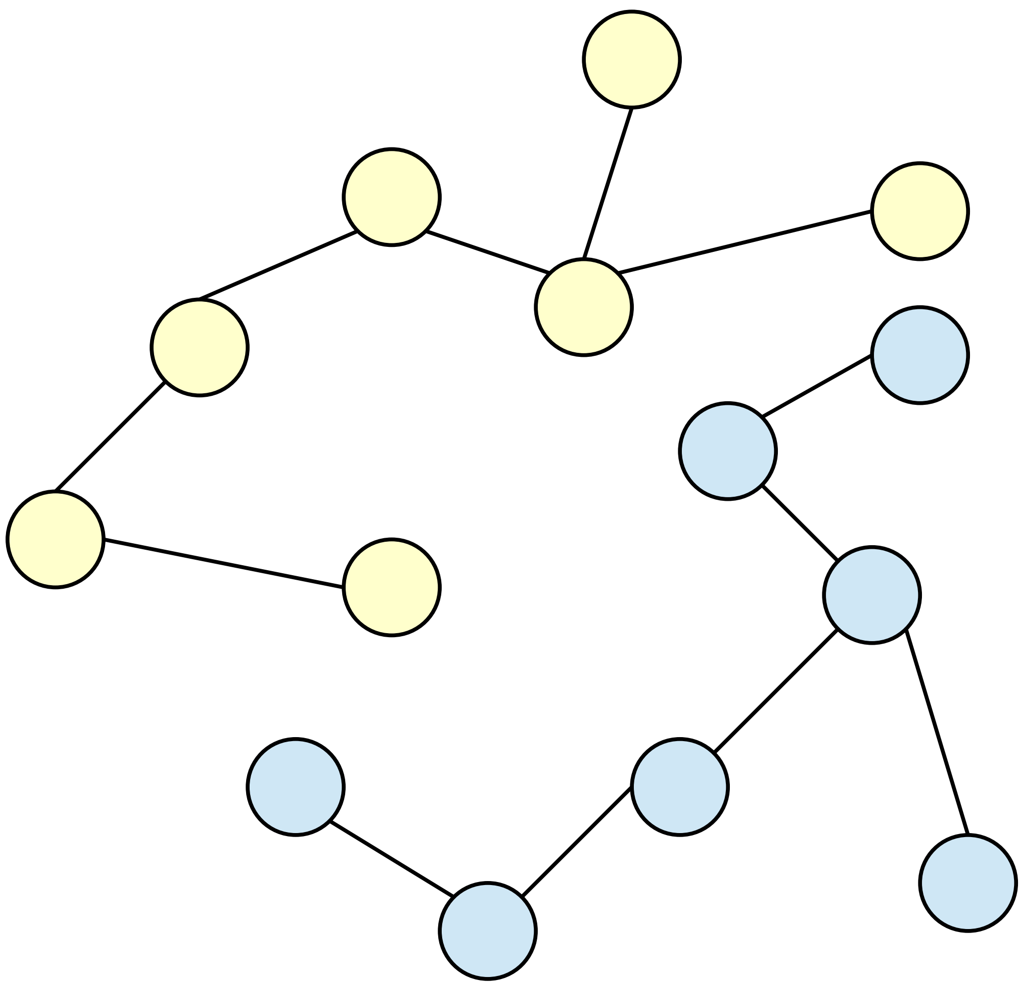


Assume any tree with at most *k* nodes has one more node than edge.

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.



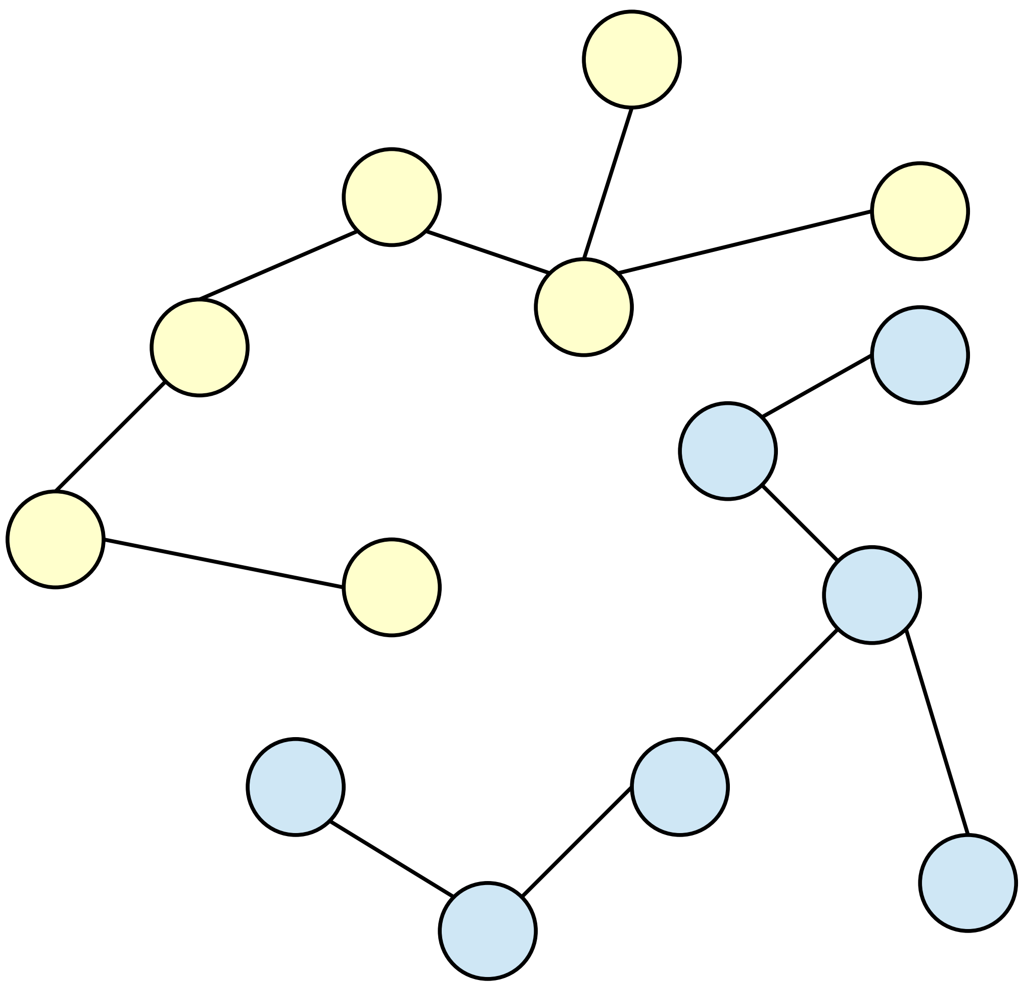
Assume any tree with at most *k* nodes has one more node than edge.

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.



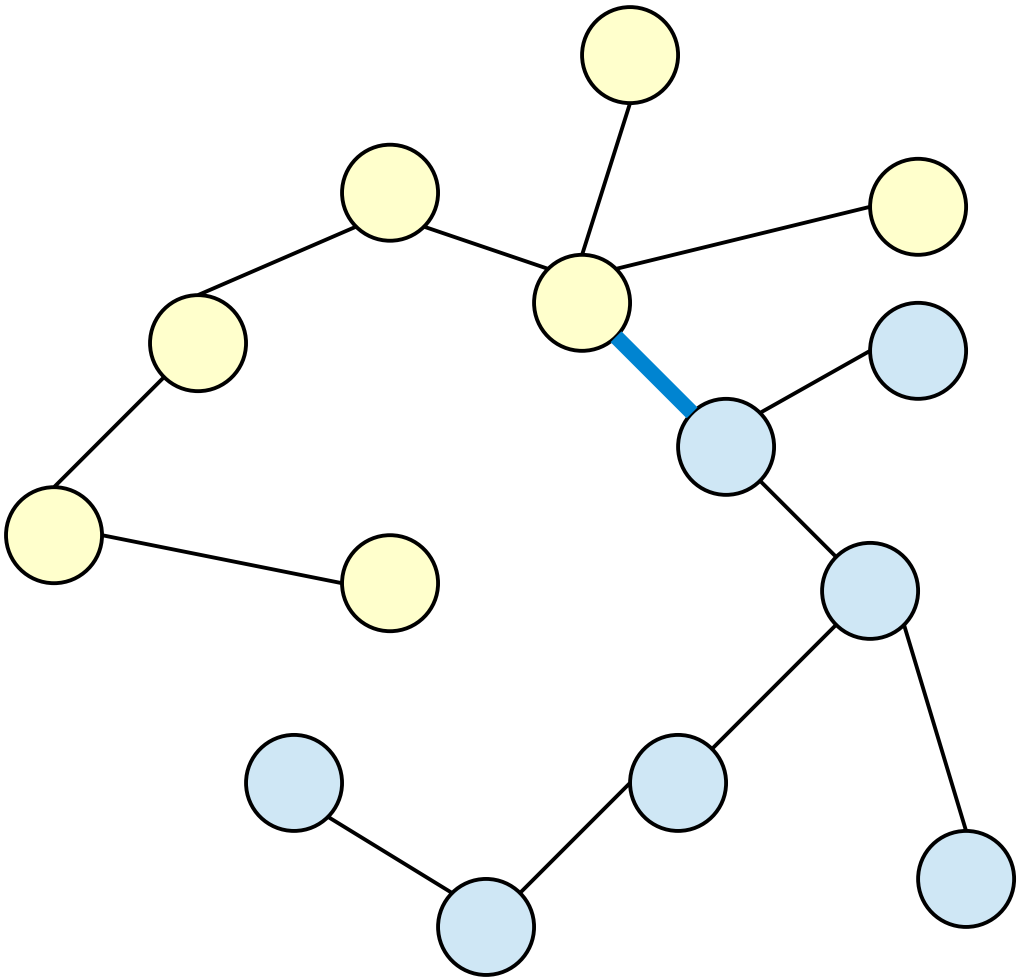
Assume any tree with at most *k* nodes has one more node than edge.

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.

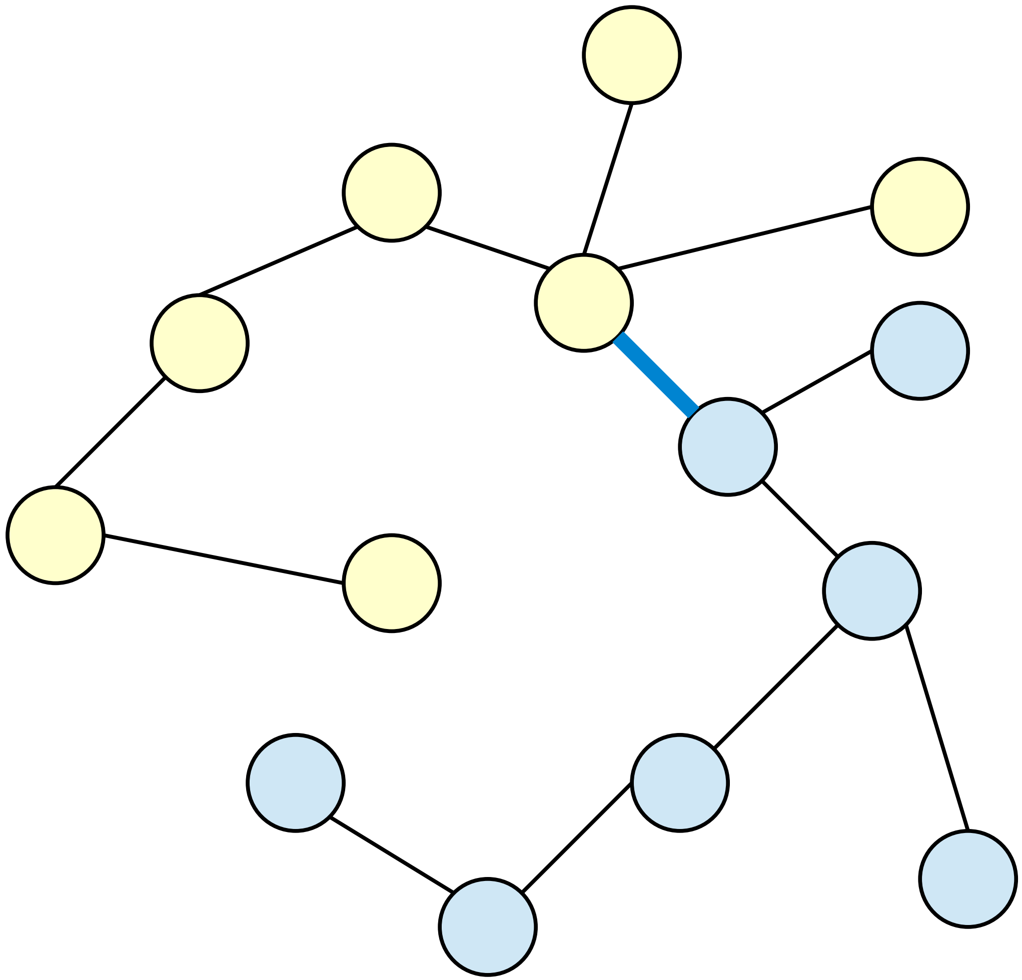
There are *r*-1 edges in the yellow tree and *k*-*r* edges in the blue tree.



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.

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."

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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 $1 + (r-1) + (k-r) = k$, 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.

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 $1 + (r-1) + (k-r) = k$, 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.

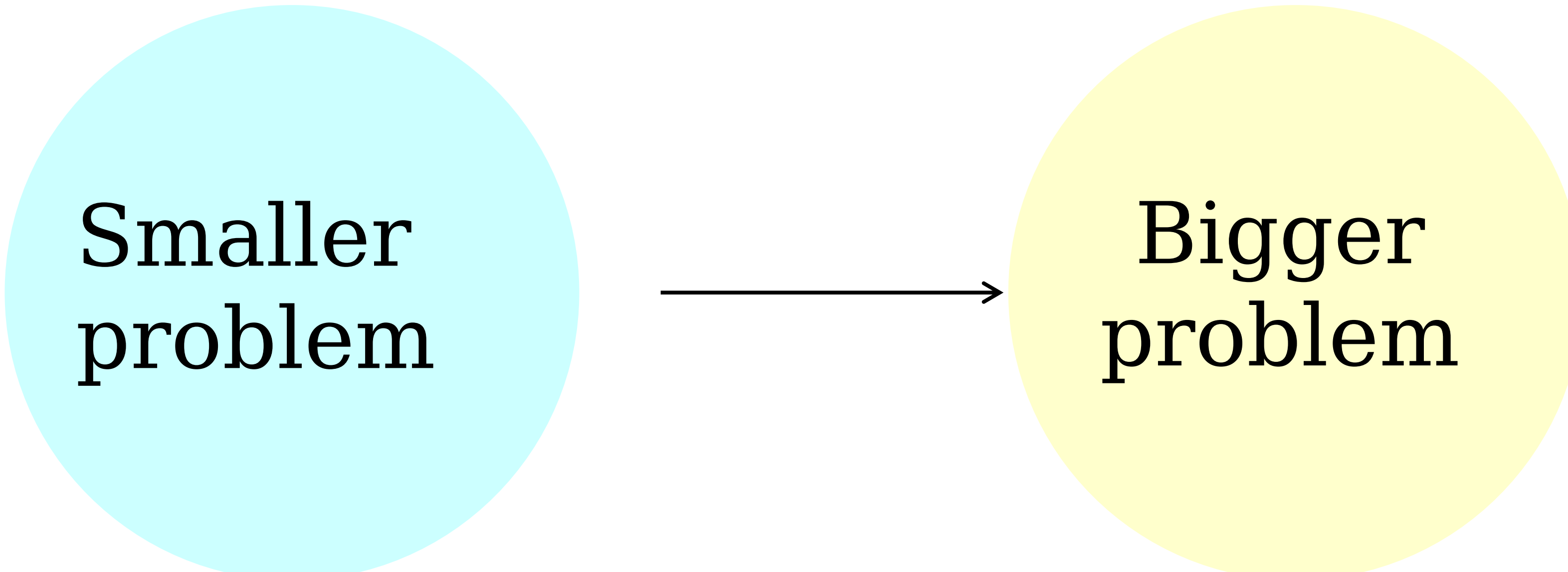
As a base case, we will prove $P(1)$, that any tree with 1 node has 0 edges. Any such tree has single node, so it cannot have any edges.

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.

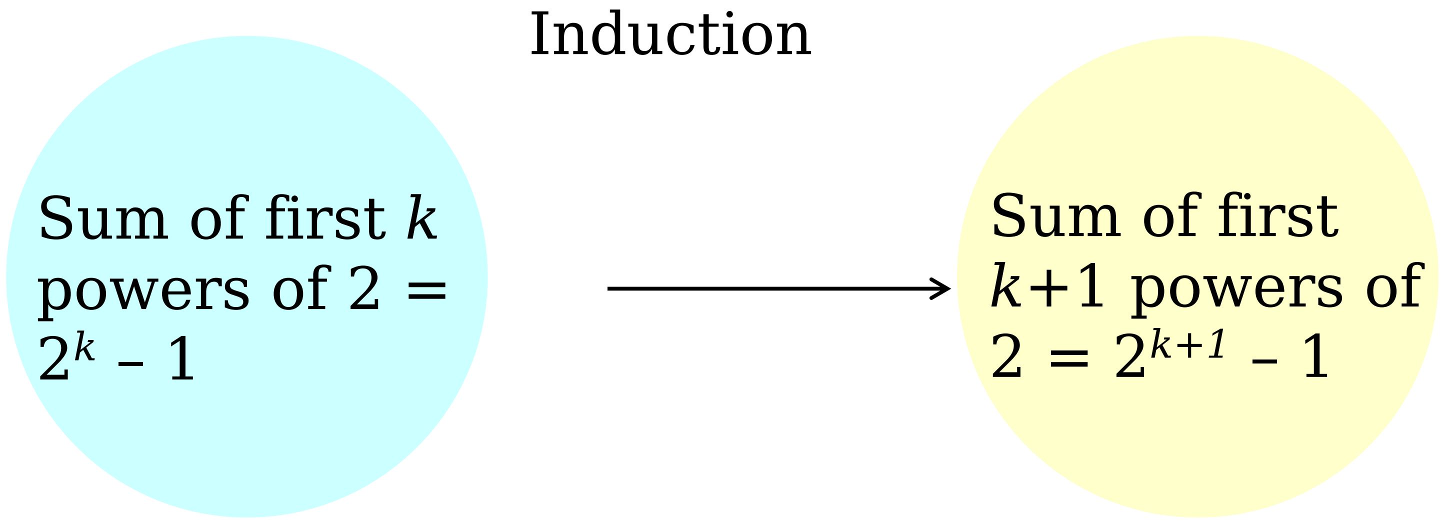
Consider any tree T with $k+1$ nodes. Since T has at least two nodes and is connected, it must contain at least one edge. Choose any edge in T and delete it. This splits T into two nonempty trees T_1 and T_2. Every edge in T is part of T_1, is part of T_2, or is the initial edge we deleted.

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. ■

Induction vs. Complete Induction



Induction vs. Complete Induction

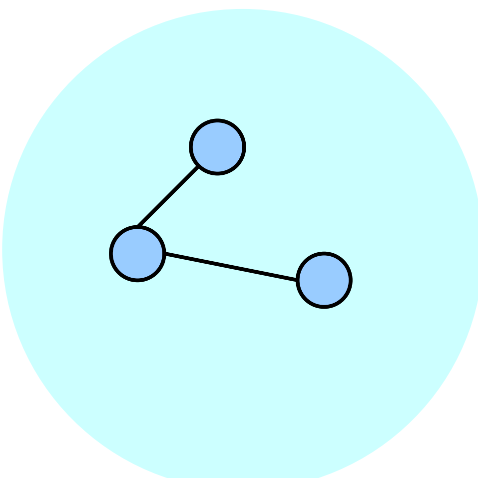
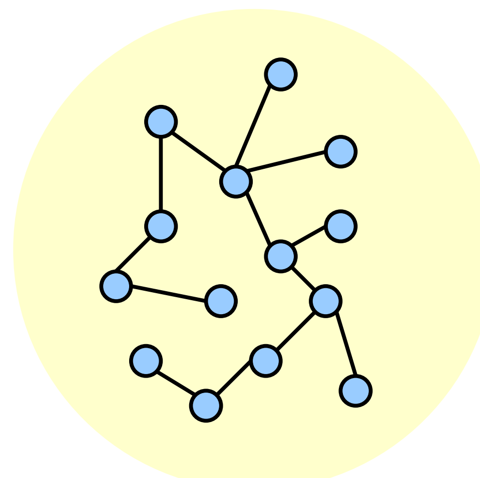


Induction vs. Complete Induction

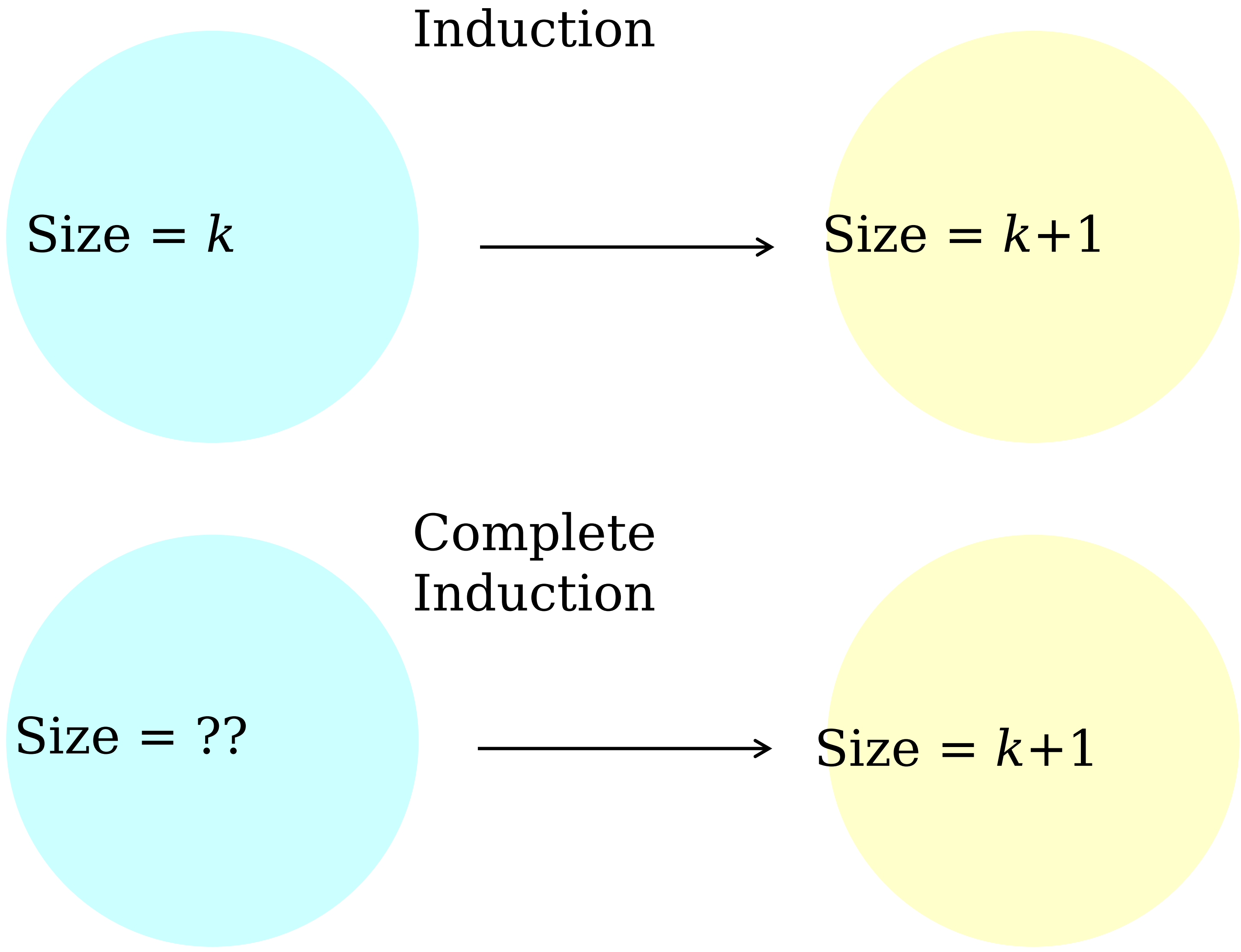
Induction

Sum of first k powers of 2 = $2^k - 1$ → Sum of first $k+1$ powers of 2 = $2^{k+1} - 1$

Complete Induction

 → 

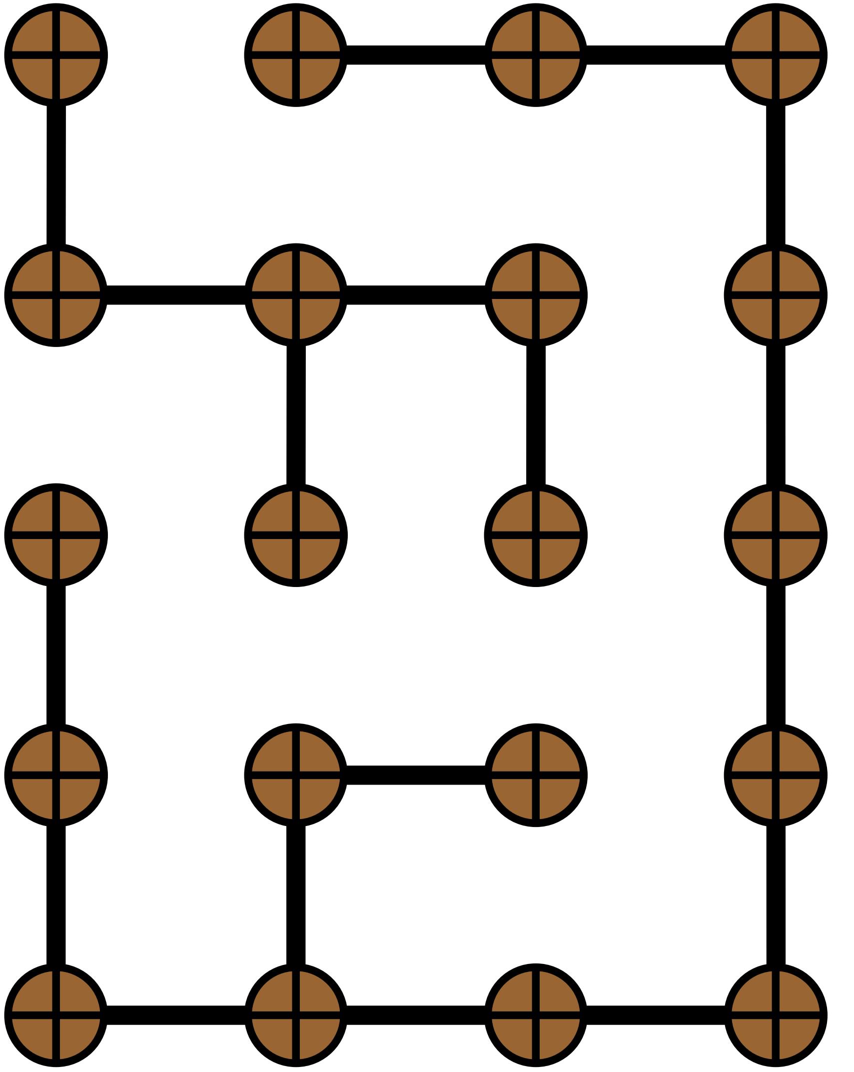
Induction vs. Complete Induction



Rat Mazes

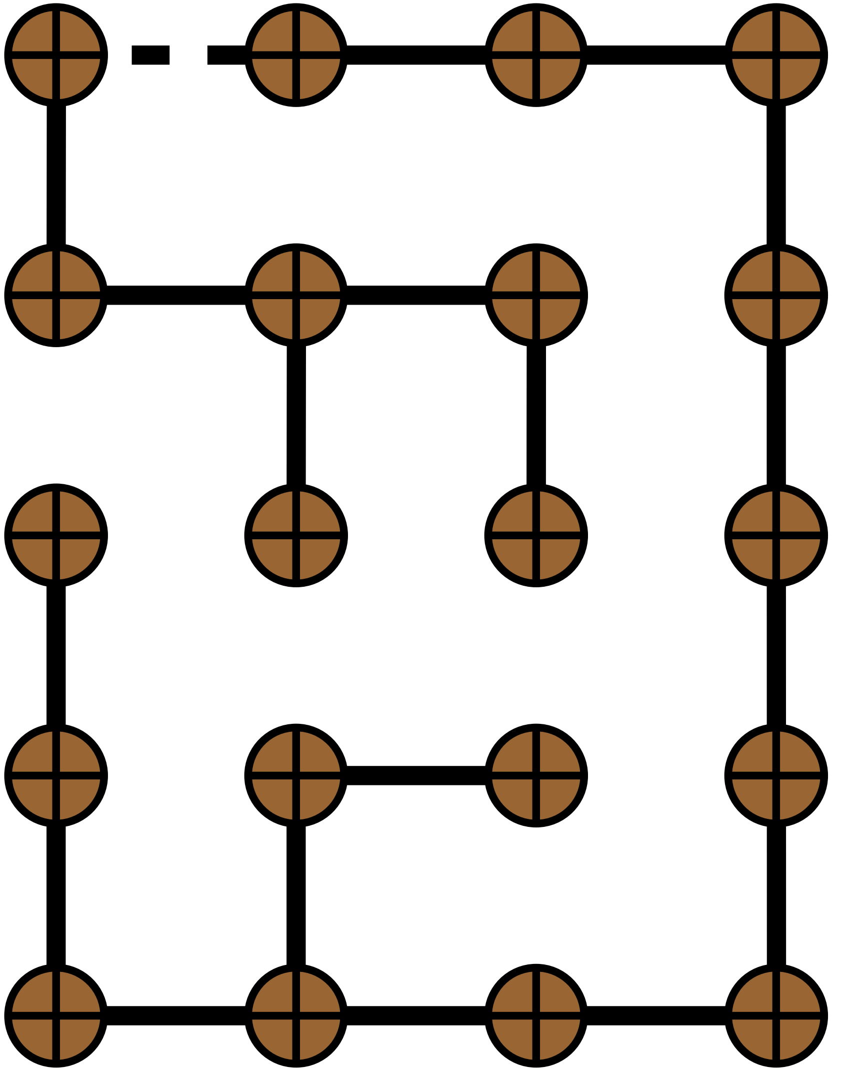
Suppose you want to make a rat maze consisting of an $n \times m$ grid of pegs with slats between them.

Question: How many slats do you need to create?



Rat Mazes

Suppose you want to make a rat maze consisting of an $n \times m$ grid of pegs with slats between them.

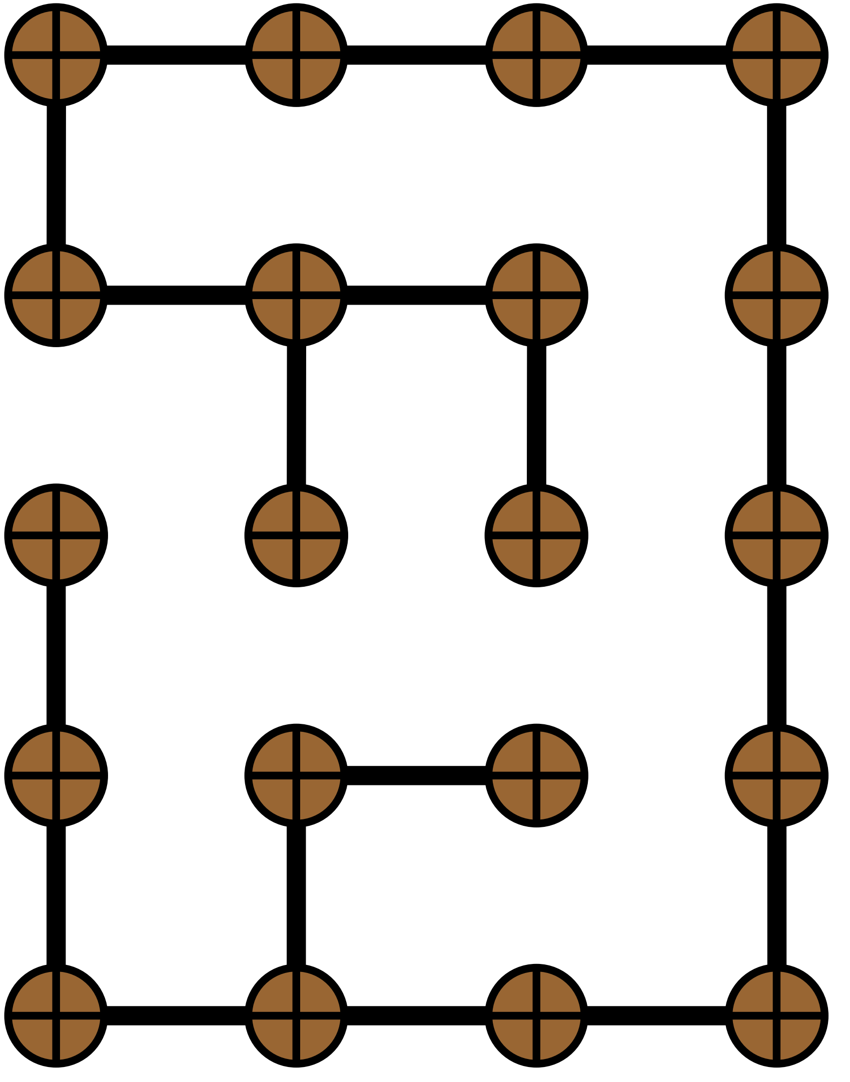


Question: How many slats do you need to create?

Rat Mazes

Suppose you want to make a rat maze consisting of an $n \times m$ grid of pegs with slats between them.

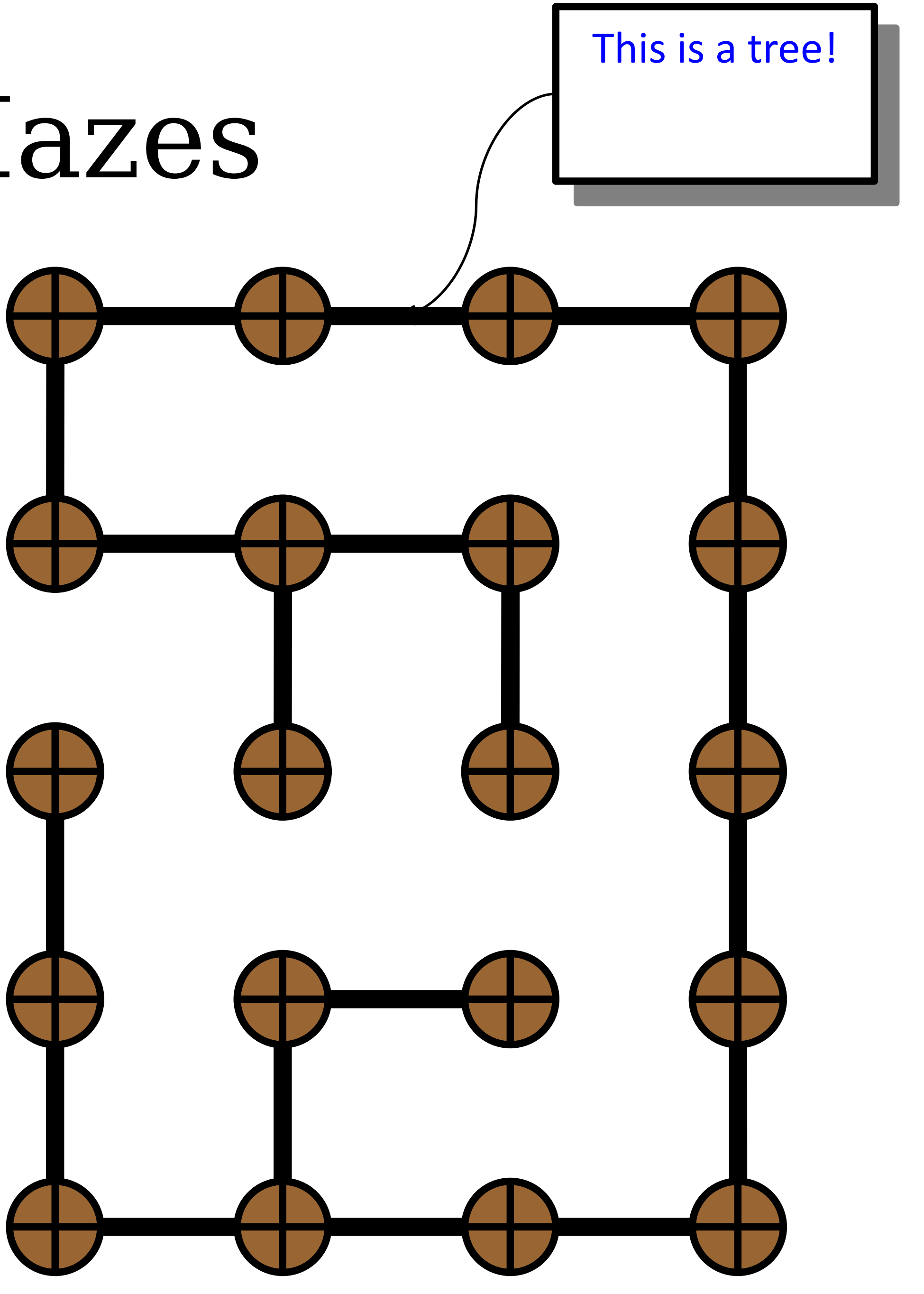
Question: How many slats do you need to create?



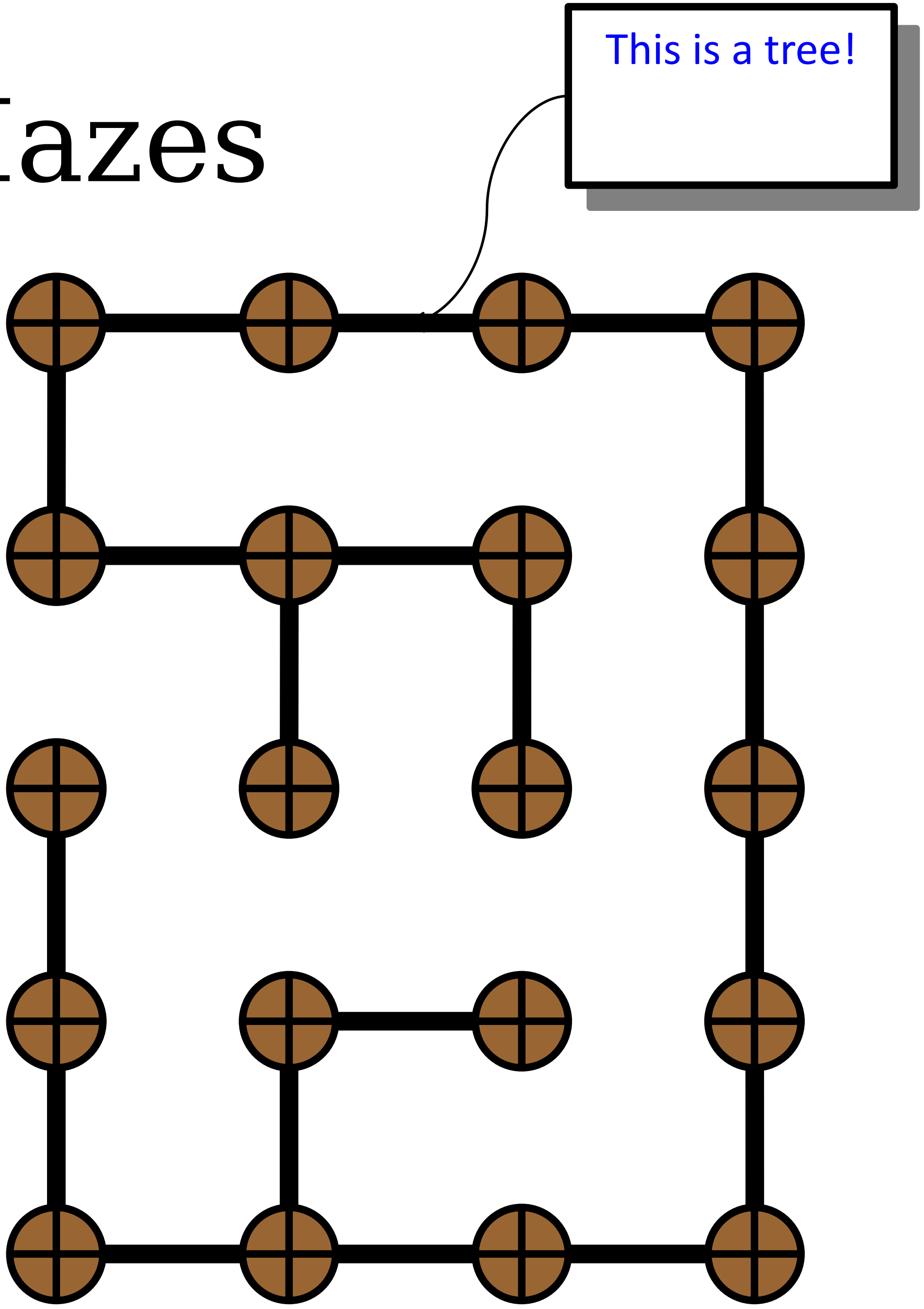
Rat Mazes

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?



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!

Three Questions

What is something you know now that, at the start of the quarter, you knew you didn’t know?

What is something you know now that, at the start of the quarter, you didn’t know that you didn’t know?

What is something you don’t know that, at the start of the quarter, you didn’t know that you didn’t know?

Let’s play a game!

Rules

Start with a pile of n coins for some $n \geq 0$

Players take turns removing between 1 and 5 coins from the pile

The player who has no more coins to remove loses the game

Rules

Start with a pile of n coins for some $n \geq 0$

Players take turns removing between 1 and 5 coins from the pile

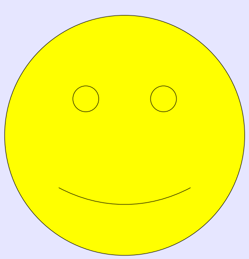
The player who has no more coins to remove loses the game

Interestingly, if the pile begins with a multiple of 6 coins in it, the second player can always win if they play correctly – give it a try!

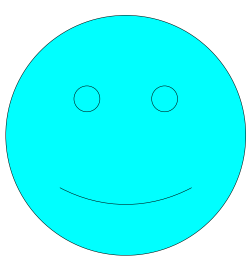
What’s the strategy?

Some Observations

If it's the first player's turn and there are no coins left, then the second player wins



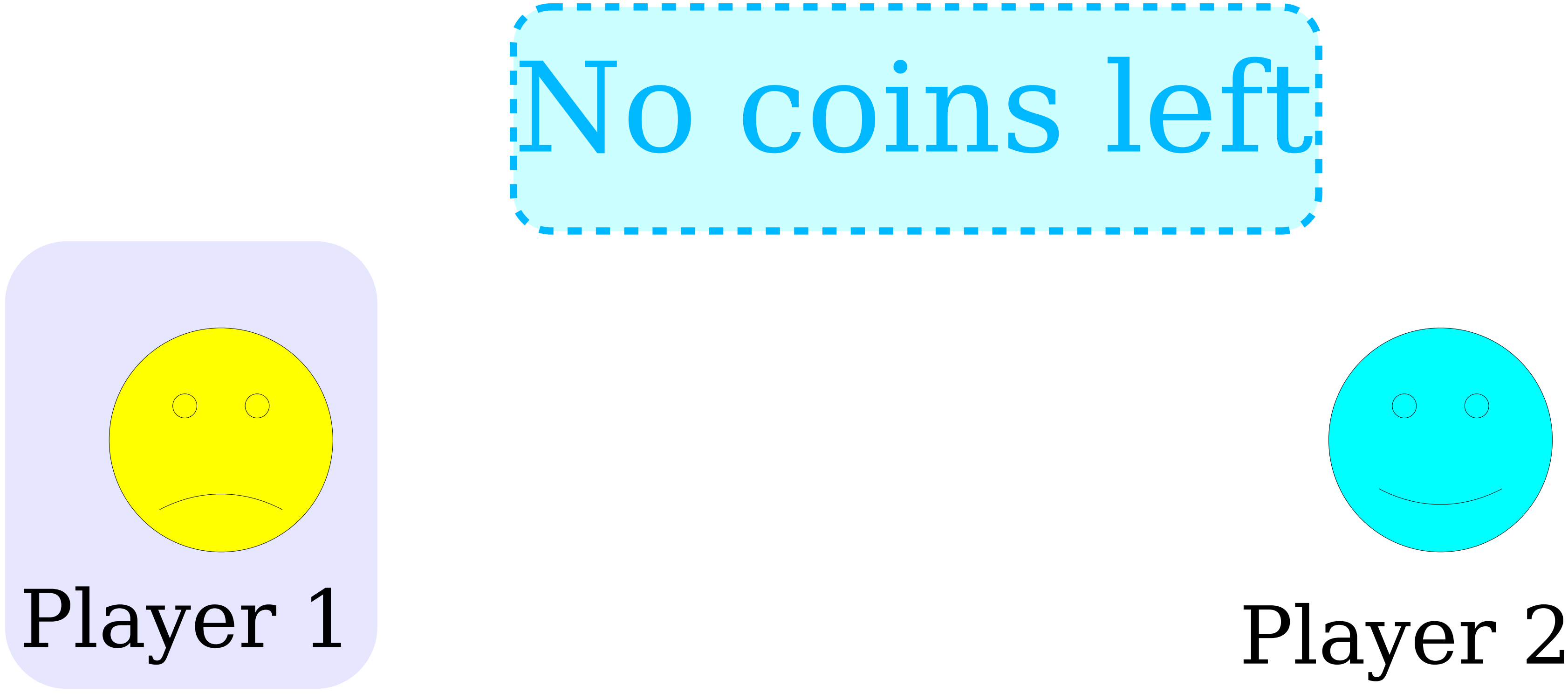
Player 1



Player 2

Some Observations

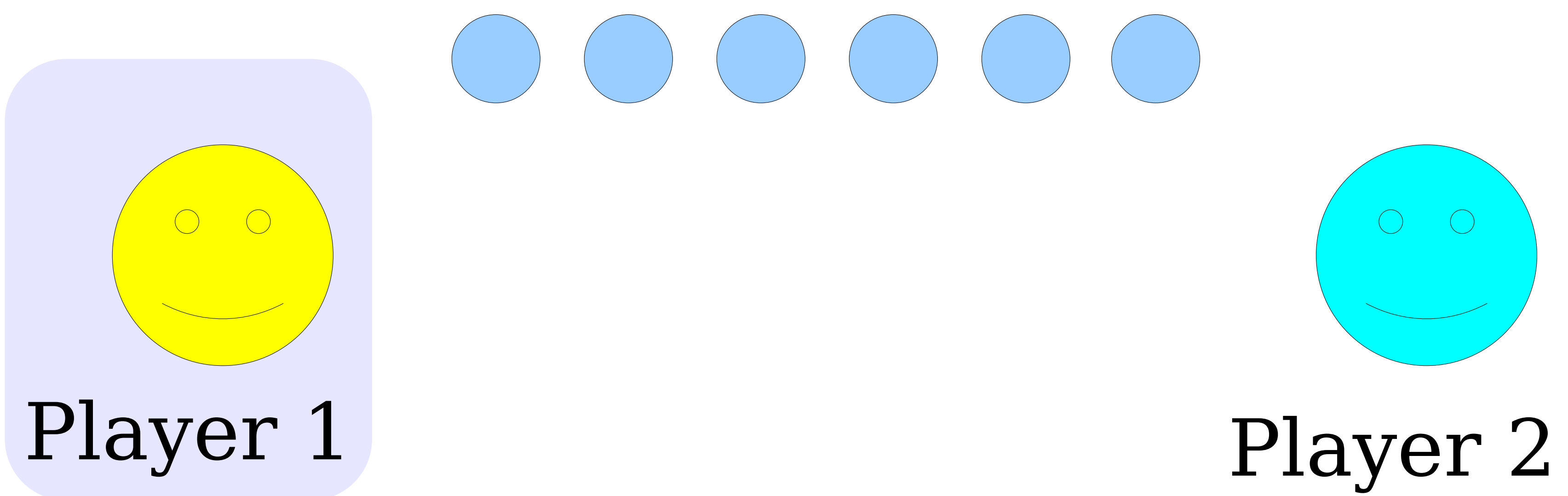
If it’s the first player’s turn and there are no coins left, then the second player wins



Some Observations

If it’s the first player’s turn and there are no coins left, then the second player wins

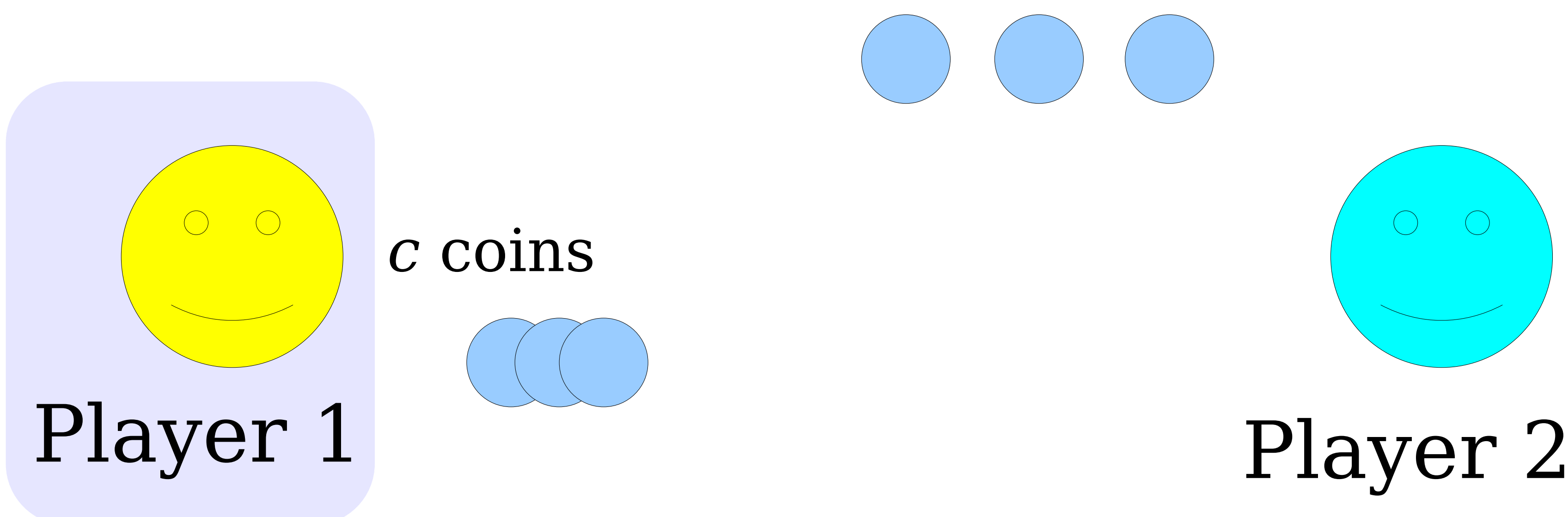
If we start with 6 coins, player 1 has to remove some but not all of the coins.



Some Observations

If it’s the first player’s turn and there are no coins left, then the second player wins

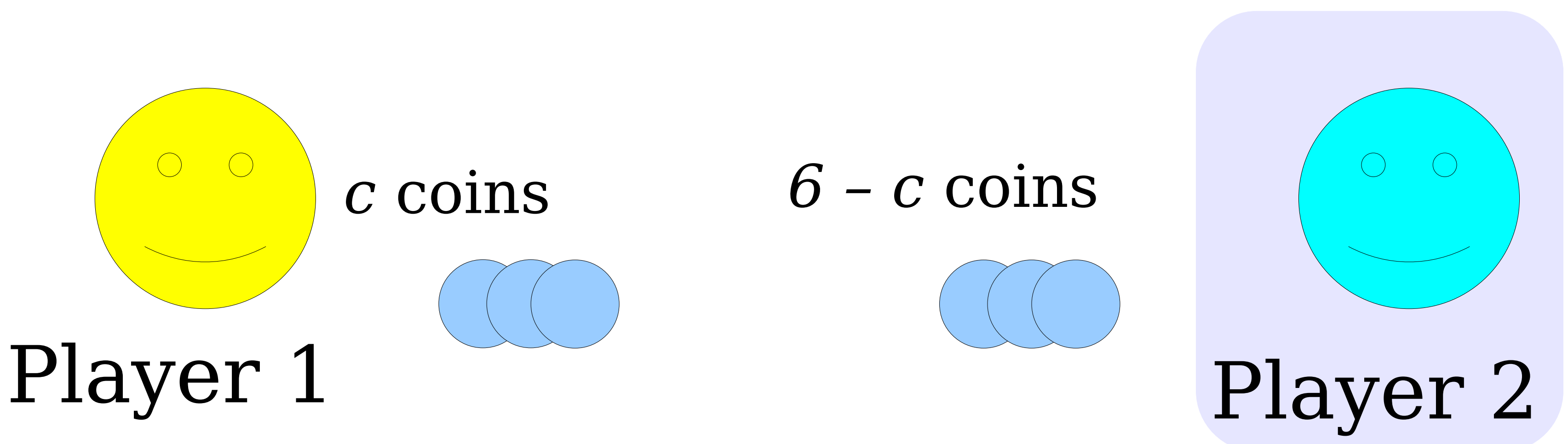
If we start with 6 coins, player 1 has to remove some but not all of the coins.



Some Observations

If it’s the first player’s turn and there are no coins left, then the second player wins

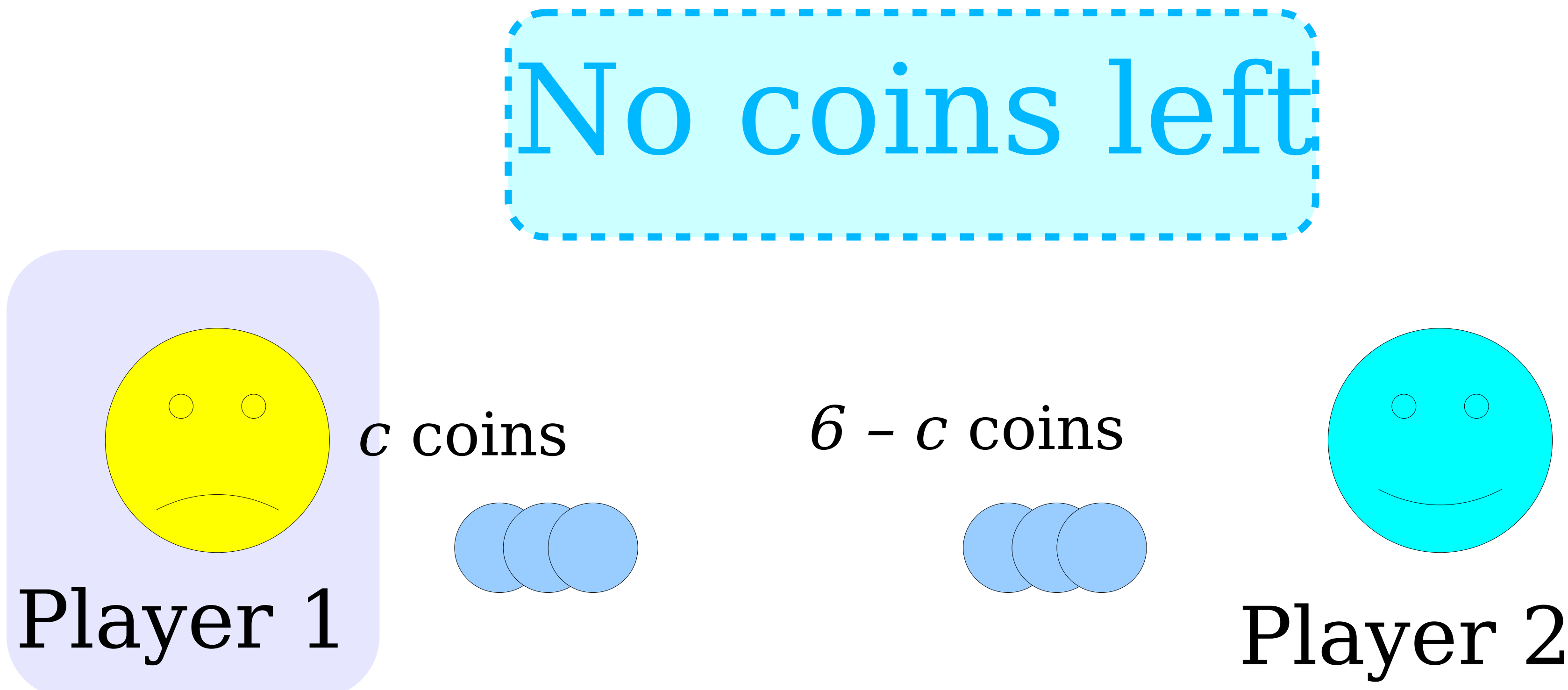
If we start with 6 coins, player 1 has to remove some but not all of the coins. Then player 2 can remove the remaining coins



Some Observations

If it’s the first player’s turn and there are no coins left, then the second player wins

If we start with 6 coins, player 1 has to remove some but not all of the coins. Then player 2 can remove the remaining coins, leaving us in a known winning state.

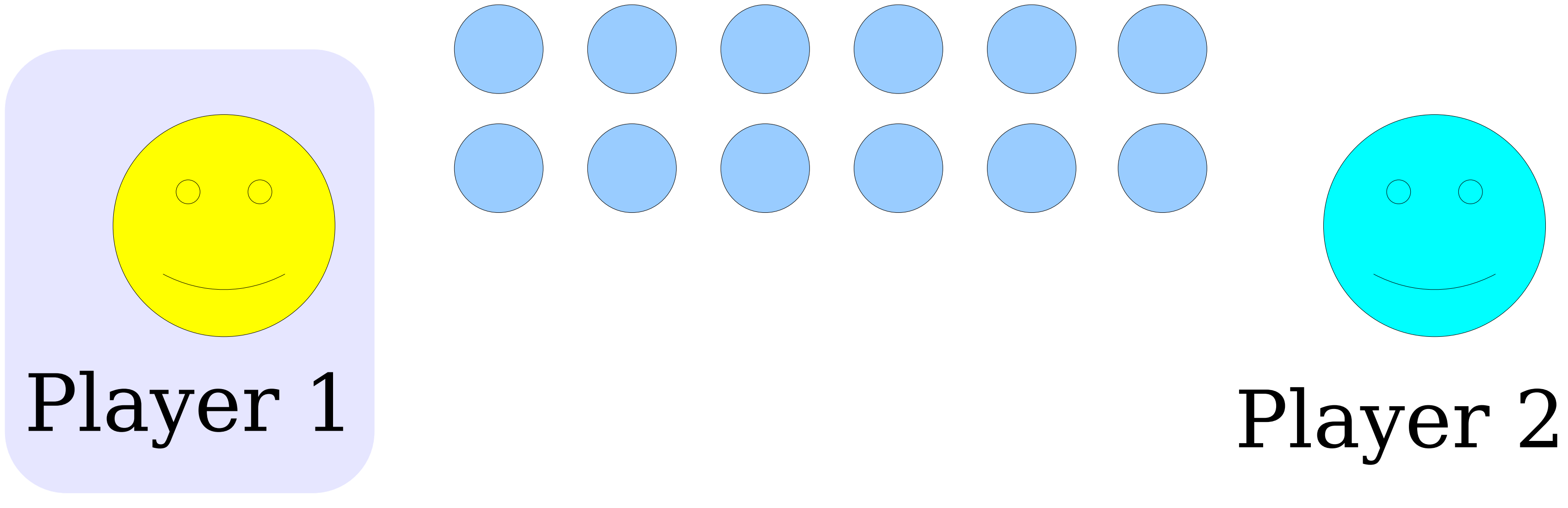


Some Observations

If it’s the first player’s turn and there are no coins left, then the second player wins

If we start with 6 coins, player 1 has to remove some but not all of the coins. Then player 2 can remove the remaining coins, leaving us in a known winning state.

What happens when there are 12 coins?

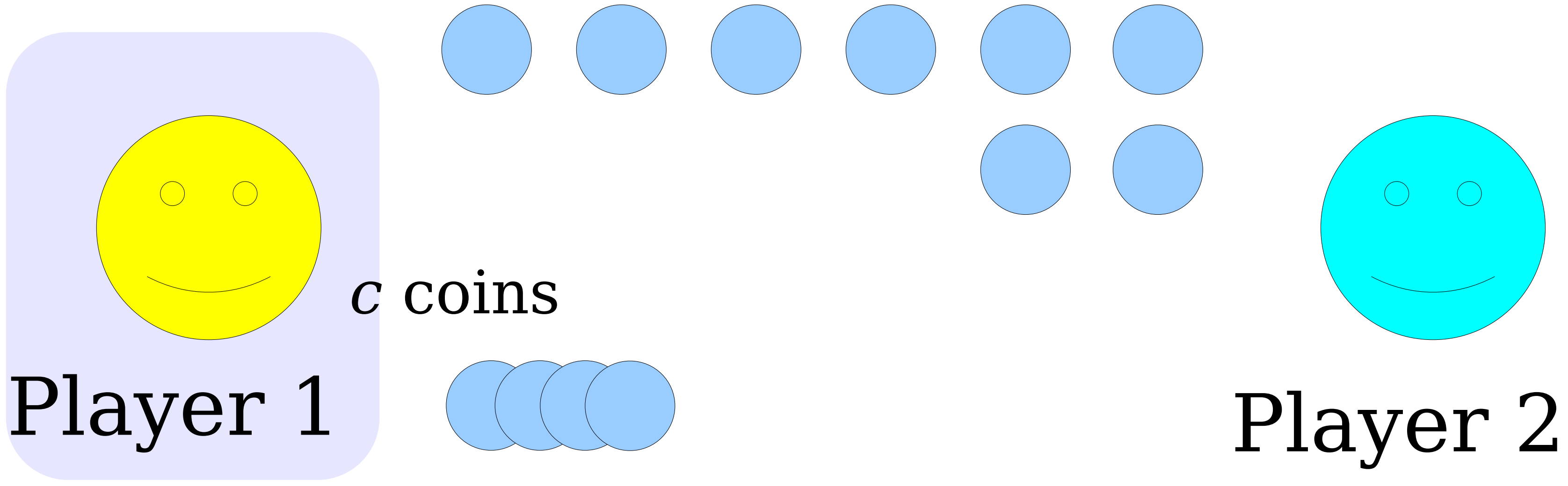


Some Observations

If it's the first player's turn and there are no coins left, then the second player wins

If we start with 6 coins, player 1 has to remove some but not all of the coins. Then player 2 can remove the remaining coins, leaving us in a known winning state.

What happens when there are 12 coins? Player 1 removes some coins

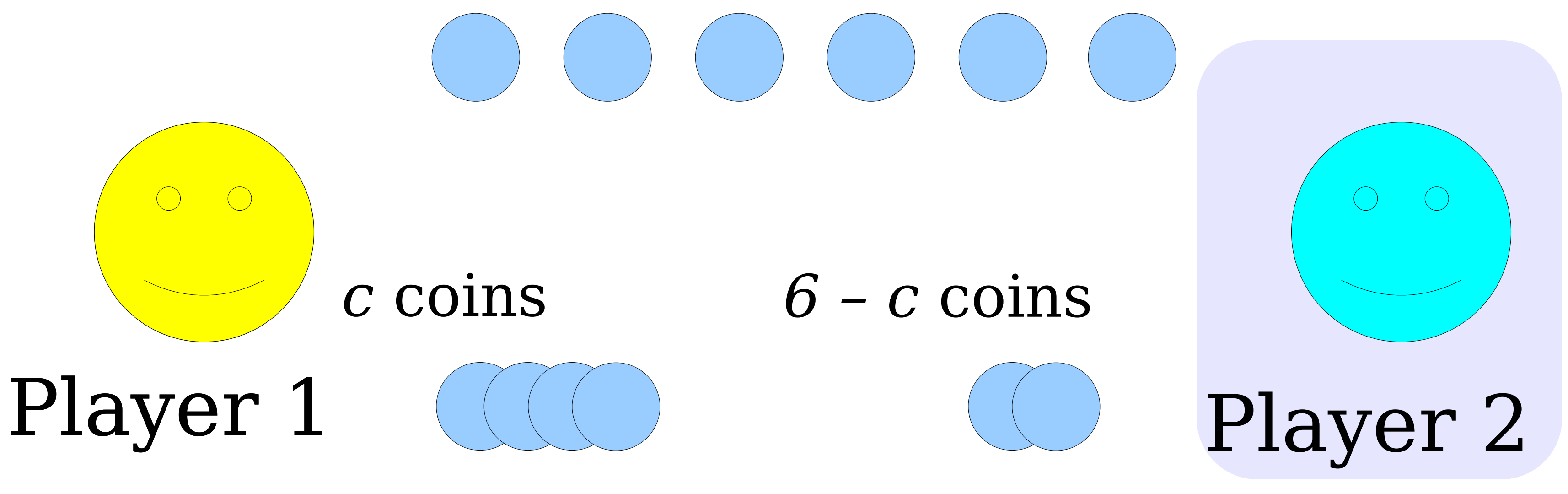


Some Observations

If it's the first player's turn and there are no coins left, then the second player wins

If we start with 6 coins, player 1 has to remove some but not all of the coins. Then player 2 can remove the remaining coins, leaving us in a known winning state.

What happens when there are 12 coins? Player 1 removes some coins, but then player 2 can remove the right number of coins to leave 6 remaining.

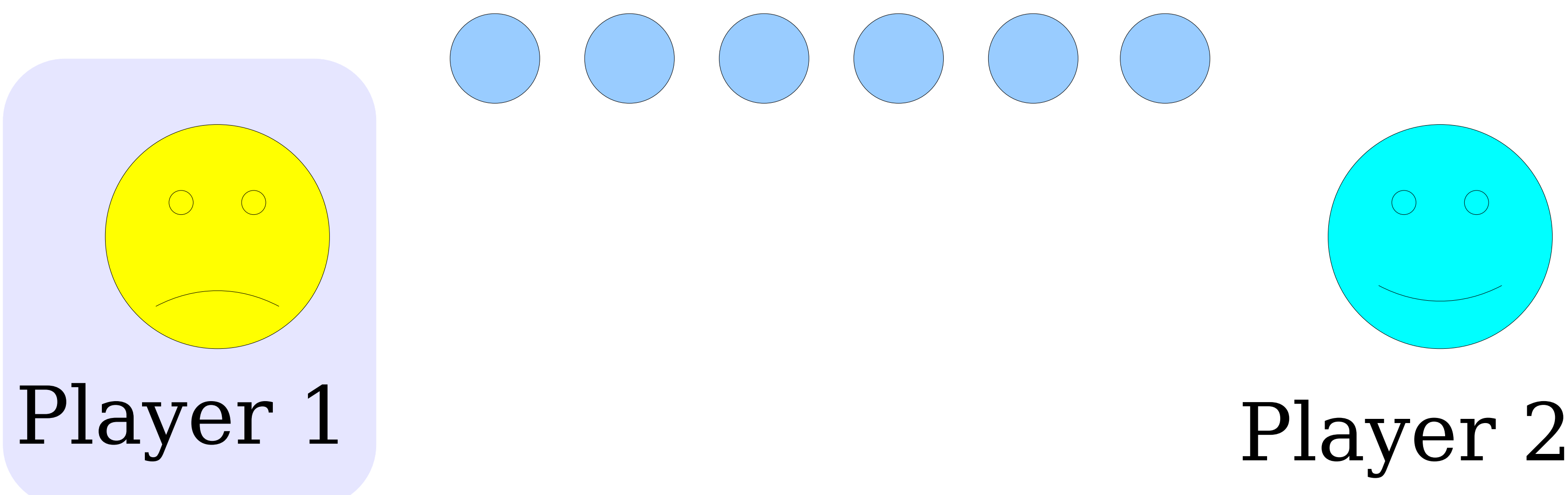


Some Observations

If it's the first player's turn and there are no coins left, then the second player wins

If we start with 6 coins, player 1 has to remove some but not all of the coins. Then player 2 can remove the remaining coins, leaving us in a known winning state.

What happens when there are 12 coins? Player 1 removes some coins, but then player 2 can remove the right number of coins to leave 6 remaining. It's player 1's turn again and there are 6 coins, again a known winning state.

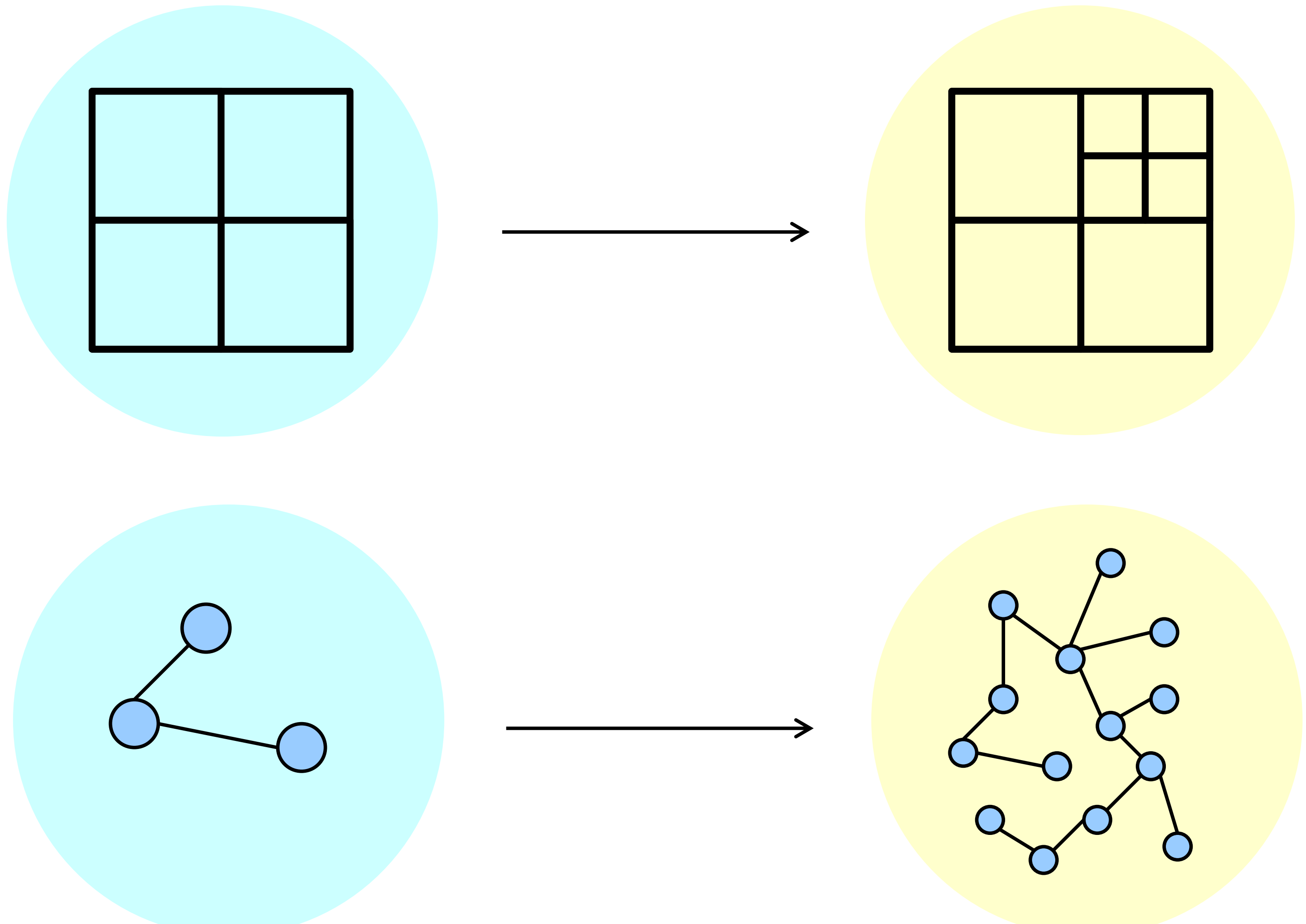


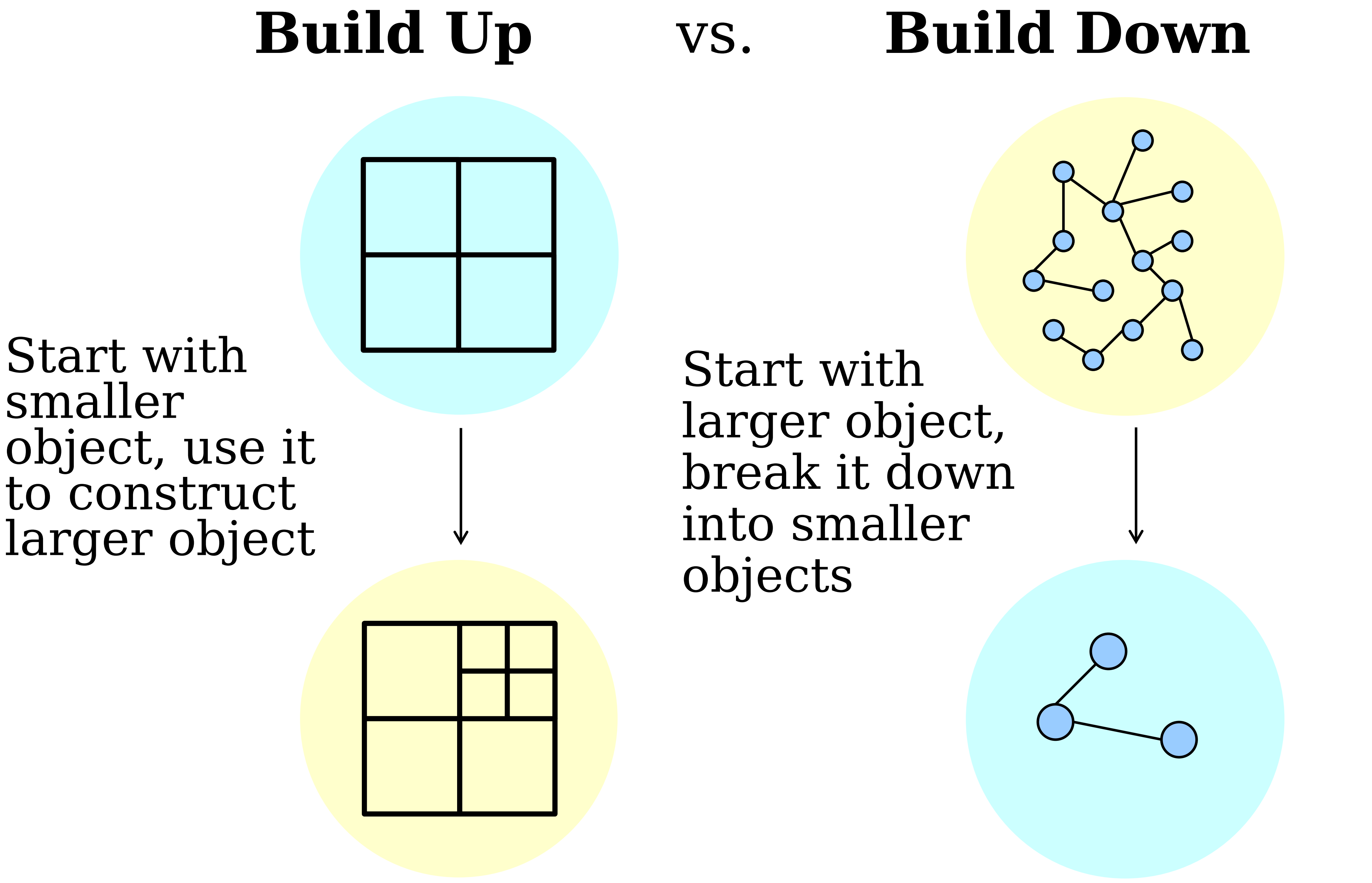
Strategy: The second player can win by making the total number of coins removed by their move and the first player’s move come out to 6.

Strategy: The second player can win by making the total number of coins removed by their move and the first player’s move come out to 6.

It is a *great* idea to *try small cases* before jumping into a formal proof. It will be much easier to formalize the logic here now that you have a feel for how to play the game.

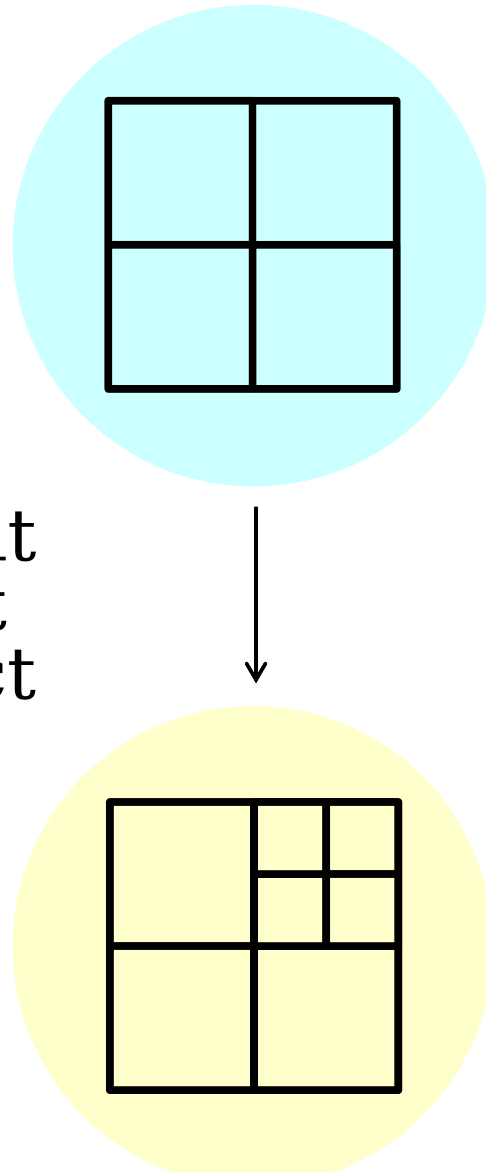
The big question in a proof by induction:
How can I leverage a smaller result to help me prove a bigger result?





Build Up
vs.
Build Down
Start with smaller object, use it to construct larger object
Start with larger object, break it down into smaller objects

Build Up



Start with smaller object, use it to construct larger object

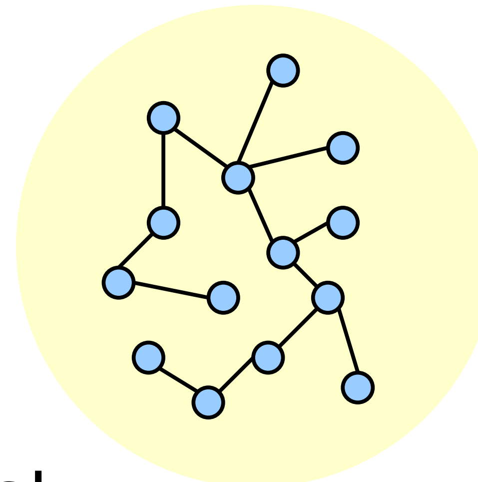
Build up if P(n) is existentially qualified

We can use our inductive hypothesis (there exists an object of size k) to prove that there exists an object of size k+1

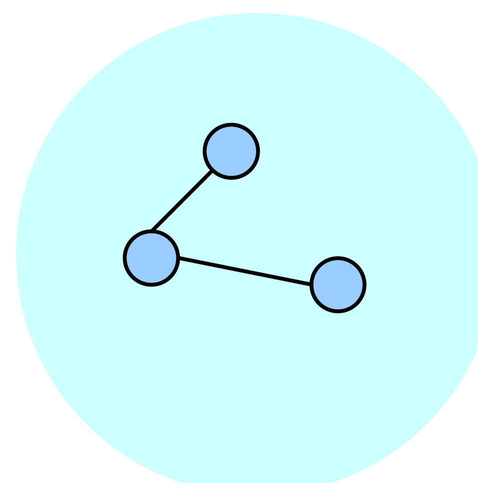
Build Down

Build down if P(n) is universally qualified

Our inductive hypothesis (for all objects of size k, some property is true) doesn't apply to an object of size k+1



Start with larger object, break it down into smaller objects



For all games where the number of coins is a multiple of 6, the second player can always win if they play correctly.

What is P(n)?

What does the problem size "n" in P(n) represent?

What is the base case?

What is the step size?

For all games where the number of coins is a multiple of 6, the second player can always win if they play correctly.

What is $P(n)$?

Let P(n) be the statement "if the game is played with the pile containing n coins, the second player can always win if she plays correctly."

What does the problem size "n" in $P(n)$ represent?

The problem size is the number of coins.

What is the base case?

The base case is n=0, the simplest possible case of the game is when you start with no coins.

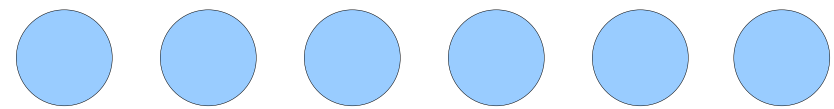
What is the step size?

We want to show the result is true for multiples of 6, so we'll take steps of size 6.

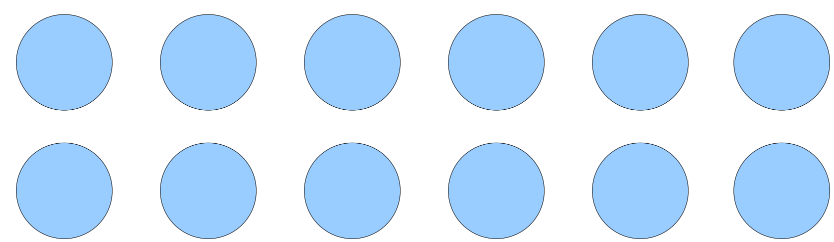
P(n) = “if the game is played with the pile containing n coins, the second player can always win if they play correctly.”

Is P(n) universally or existentially quantified? Based on that, should we build up or build down?

P(k)



P(k+6)

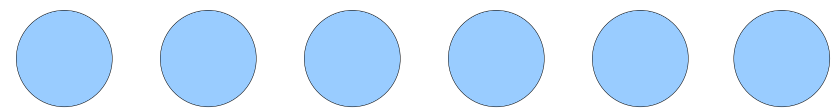


P(n) = “if the game is played with the pile containing n coins, the second player can always win if they play correctly.”

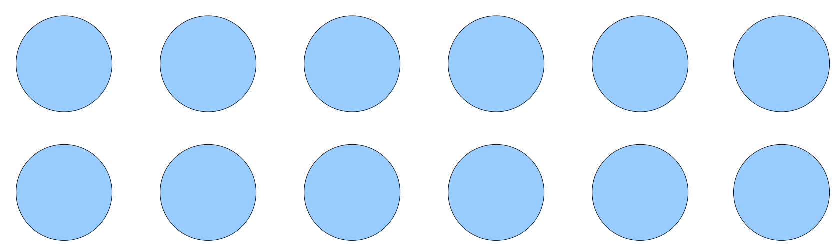
Is P(n) universally or existentially quantified? Based on that, should we build up or build down?

P(n) is universally qualified so we should build down (start with a game of size k+6 and figure out how to reduce it to a game of size k).

P(k)



P(k+6)

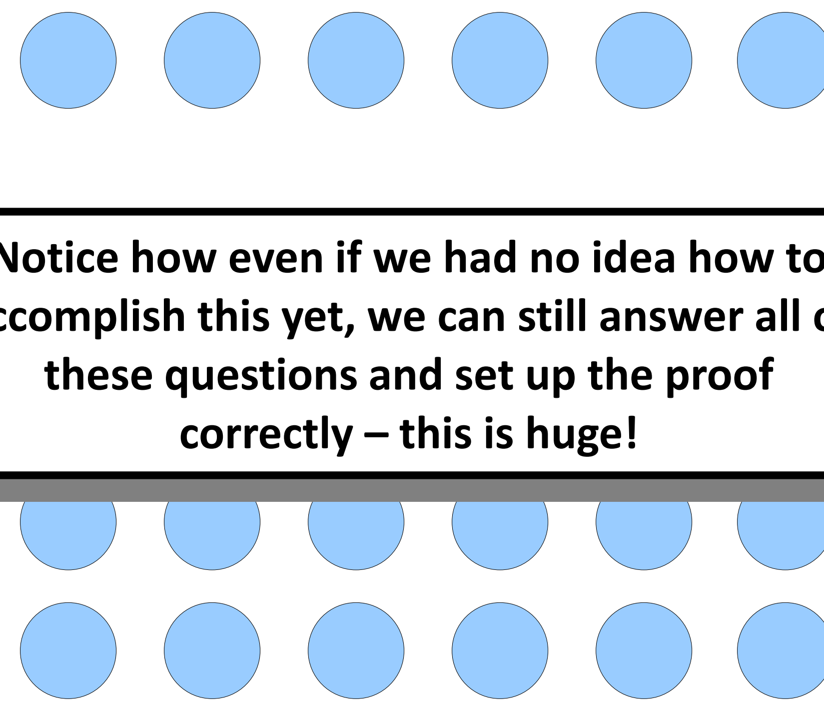


P(n) = "if the game is played with the pile containing n coins, the second player can always win if they play correctly."

Is P(n) universally or existentially quantified? Based on that, should we build up or build down?

P(n) is universally qualified so we should build down (start with a game of size k+6 and figure out how to reduce it to a game of size k).

P(k)

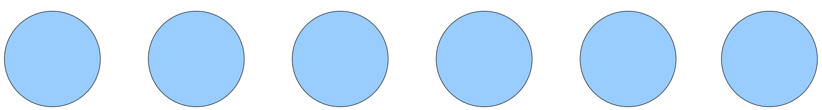


Notice how even if we had no idea how to accomplish this yet, we can still answer all of these questions and set up the proof correctly – this is huge!

P(n) = “if the game is played with the pile containing n coins, the second player can always win if they play correctly.”

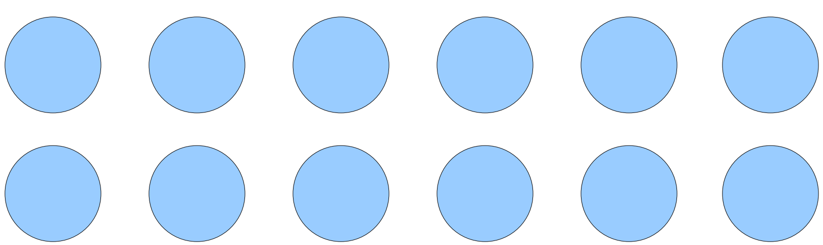
Assume P(k)

(If the game is played with k coins, the second player can always win)



Prove P(k+6)

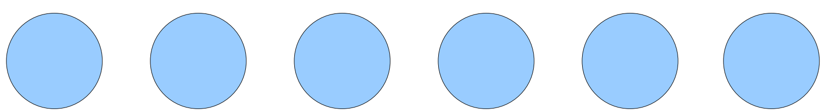
(If the game is played with k+6 coins, the second player can always win)



P(n) = "if the game is played with the pile containing n coins, the second player can always win if they play correctly."

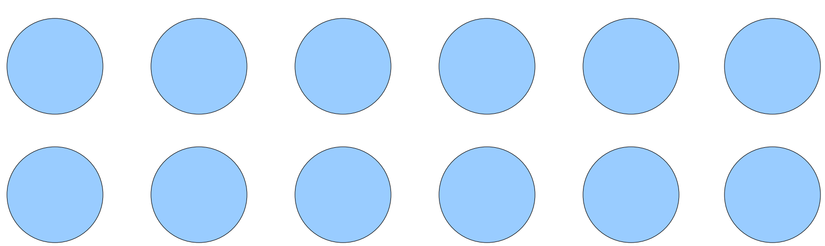
Assume P(k)

(If the game is played with k coins, the second player can always win)



Prove P(k+6)

(If the game is played with k+6 coins, the second player can always win)

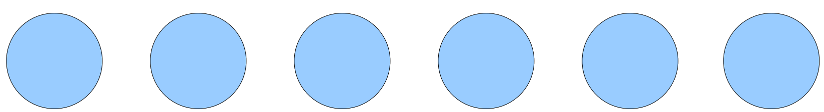


We want to take a game with k+6 coins and explain the strategy for reducing that game into one with just *k coins* so that we can apply the inductive hypothesis

P(n) = "if the game is played with the pile containing n coins, the second player can always win if they play correctly."

Assume P(k)

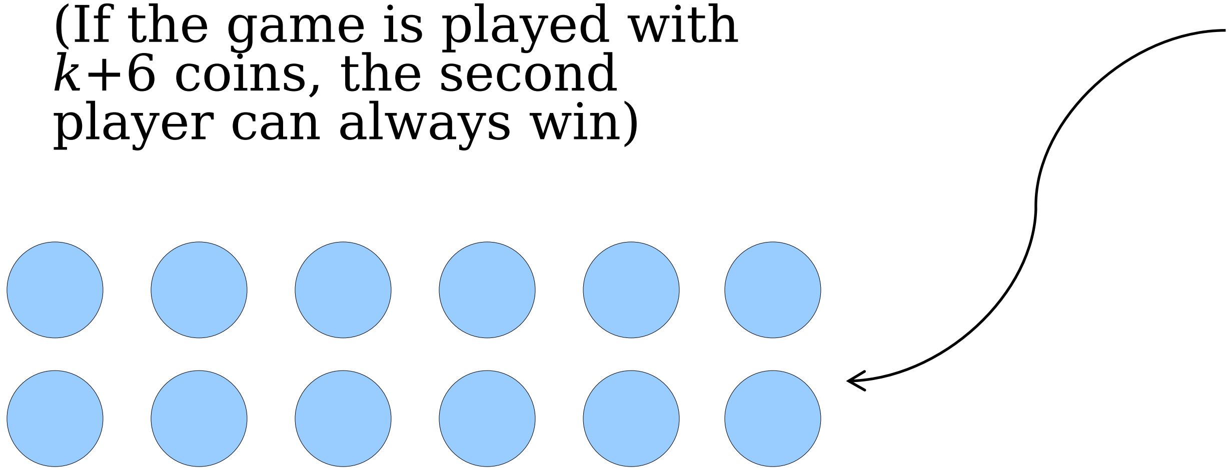
(If the game is played with k coins, the second player can always win)



Prove P(k+6)

(If the game is played with k+6 coins, the second player can always win)

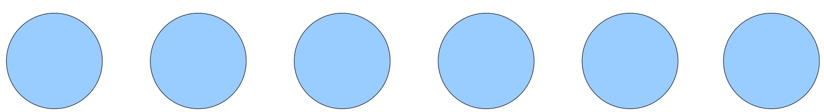
Strategy: The second player can win by making the total number of coins removed by their move and the first player's move come out to 6.



P(n) = “if the game is played with the pile containing n coins, the second player can always win if they play correctly.”

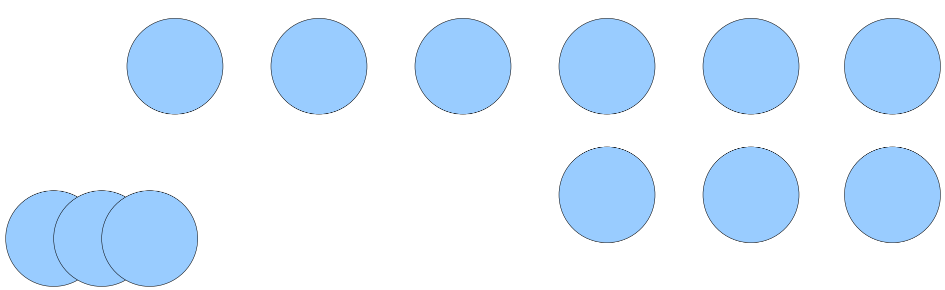
Assume P(k)

(If the game is played with k coins, the second player can always win)



Prove P(k+6)

(If the game is played with k+6 coins, the second player can always win)

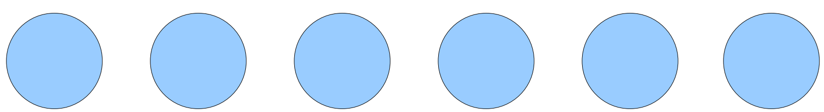


Strategy: The second player can win by making the total number of coins removed by their move and the first player’s move come out to 6.

P(n) = “if the game is played with the pile containing n coins, the second player can always win if they play correctly.”

Assume P(k)

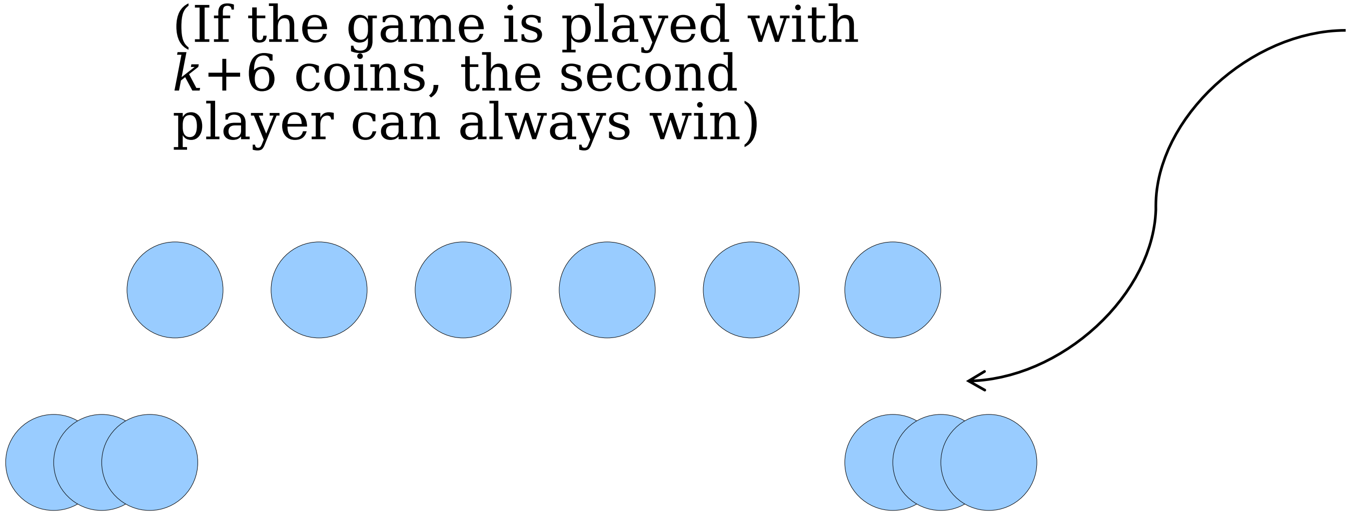
(If the game is played with k coins, the second player can always win)



Prove P(k+6)

(If the game is played with k+6 coins, the second player can always win)

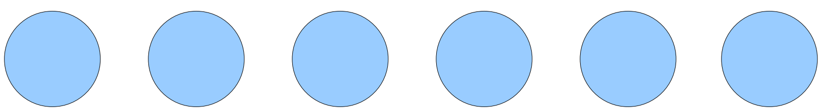
Strategy: The second player can win by making the total number of coins removed by their move and the first player’s move come out to 6.



P(n) = “if the game is played with the pile containing n coins, the second player can always win if they play correctly.”

Assume P(k)

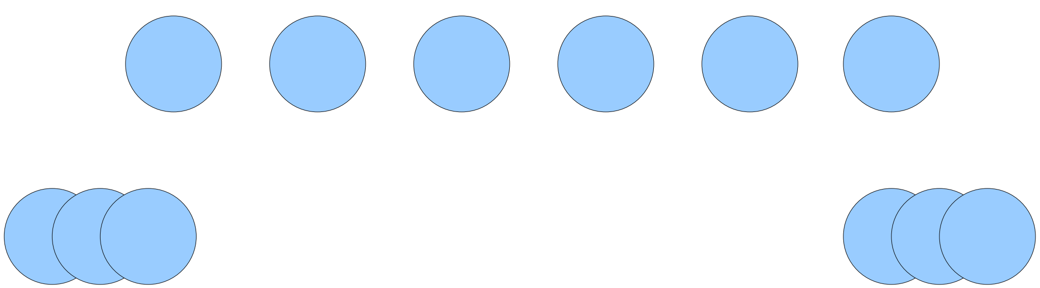
(If the game is played with k coins, the second player can always win)



Now there are k coins remaining, meaning that we can apply our inductive hypothesis

Prove P(k+6)

(If the game is played with k+6 coins, the second player can always win)



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

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.