Sequences
Counting ancestors

Problem: count number of ancestors

_ one has 2 parents, 4 grandparents, 8 great-grandparents, …, written in a row as_ 

2, 4, 8, 16, 32, 64, 128,…

Pattern?

<table>
<thead>
<tr>
<th>Position in the row</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7…</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of ancestors</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>16</td>
<td>32</td>
<td>64</td>
<td>128…</td>
</tr>
</tbody>
</table>
Sequences

Problem: count number of ancestors

one has 2 parents, 4 grandparents, 8 great-grandparents, …, written in a row as

2, 4, 8, 16, 32, 64, 128,…

Pattern?

<table>
<thead>
<tr>
<th>Position in the row</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7…</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of ancestors</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>16</td>
<td>32</td>
<td>64</td>
<td>128…</td>
</tr>
</tbody>
</table>

For a general value of $k$, $A_k$ denote number of ancestors in $k$-th generation back:

$$A_k = 2^k.$$
Sequences

In a sequence $a_1, a_2, a_3, \ldots, a_k, \ldots$

- each individual element $a_k$ ("a sub k") is called a **term**.
- $k$ in $a_k$ is called a **subscript** or **index**
- An **explicit formula** or **closed formula** for a sequence is a rule/formula that shows how value of $a_k$ depends on $k$.

\[ a_k = \frac{k}{k+1} \quad \text{for all integers } k \geq 1, \]

\[ b_i = \frac{i-1}{i} \quad \text{for all integers } i \geq 2. \]
Summation Notation

What is total number of ancestors for past six generations?

The answer is

\[ A_1 + A_2 + A_3 + A_4 + A_5 + A_6 = 2^1 + 2^2 + 2^3 + 2^4 + 2^5 + 2^6 = 126. \]

need a shorthand notation to write such sums.
Summation Notation

**Definition**

If $m$ and $n$ are integers and $m \leq n$, the symbol $\sum_{k=m}^{n} a_k$, read the summation from $k$ equals $m$ to $n$ of $a$-sub-$k$, is the sum of all the terms $a_m, a_{m+1}, a_{m+2}, \ldots, a_n$. We say that $a_m + a_{m+1} + a_{m+2} + \ldots + a_n$ is the **expanded form** of the sum, and we write

$$\sum_{k=m}^{n} a_k = a_m + a_{m+1} + a_{m+2} + \ldots + a_n.$$ 

We call $k$ the **index** of the summation, $m$ the **lower limit** of the summation, and $n$ the **upper limit** of the summation.

*introduced in 1772 by French mathematician Joseph Louis Lagrange*
Terms of a summation are often expressed using its explicit formula.

e.g.,

\[ \sum_{k=1}^{5} k^2 \quad \text{or} \quad \sum_{i=0}^{8} \frac{(-1)^i}{i + 1}. \]

Example: Expand summation form:

\[ \sum_{i=0}^{n} \frac{(-1)^i}{i + 1}. \]
Express following using summation notation:

$$\frac{1}{n} + \frac{2}{n+1} + \frac{3}{n+2} + \cdots + \frac{n+1}{2n}.$$
If \( m \) is any integer, then

\[
\sum_{k=m}^{m} a_k = a_m \quad \text{and} \quad \sum_{k=m}^{n} a_k = \sum_{k=m}^{n-1} a_k + a_n \quad \text{for all integers } n > m.
\]
Separating Off a Final Term and Adding On a Final Term

Useful manipulation of a summation
• separate off the final term of a summation

\[
\sum_{i=1}^{n+1} \frac{1}{i^2}
\]

• add/obsorbing a final term to a summation.

Write \( \sum_{k=0}^{n} 2^k + 2^{n+1} \) as a single summation.
A Telescoping Sum

Some sums can be transformed into telescoping sums, which then can be rewritten as a simple expression.

e.g., observe that

\[
\frac{1}{k} - \frac{1}{k+1} = \frac{(k+1) - k}{k(k+1)} = \frac{1}{k(k+1)}.
\]

Now we can evaluate

\[
\sum_{k=1}^{n} \frac{1}{k(k+1)}.
\]
Product Notation

The Greek capital letter pi, $\Pi$, denotes a product.

\[
\prod_{k=1}^{5} a_k = a_1 a_2 a_3 a_4 a_5.
\]

**Definition**

If $m$ and $n$ are integers and $m \leq n$, the symbol $\prod_{k=m}^{n} a_k$, read the **product from** $k$ **equals** $m$ **to** $n$ **of** $a$-sub-$k$, is the product of all the terms $a_m$, $a_{m+1}$, $a_{m+2}$, \ldots, $a_n$.

We write

\[
\prod_{k=m}^{n} a_k = a_m \cdot a_{m+1} \cdot a_{m+2} \cdots a_n.
\]
A recursive definition for the product notation is the following: If $m$ is any integer, then

$$\prod_{k=m}^{m} a_k = a_m$$

and

$$\prod_{k=m}^{n} a_k = \left(\prod_{k=m}^{n-1} a_k\right) \cdot a_n$$

for all integers $n > m$. 
Example 11 – Computing Products

Compute the following products:

a. \[ \prod_{k=1}^{5} k \]

b. \[ \prod_{k=1}^{1} \frac{k}{k+1} \]
Properties of Summations and Products

**Theorem 5.1.1**

If $a_m, a_{m+1}, a_{m+2}, \ldots$ and $b_m, b_{m+1}, b_{m+2}, \ldots$ are sequences of real numbers and $c$ is any real number, then the following equations hold for any integer $n \geq m$:

1. \[ \sum_{k=m}^{n} a_k + \sum_{k=m}^{n} b_k = \sum_{k=m}^{n} (a_k + b_k) \]

2. \[ c \cdot \sum_{k=m}^{n} a_k = \sum_{k=m}^{n} c \cdot a_k \quad \text{generalized distributive law} \]

3. \[ \left( \prod_{k=m}^{n} a_k \right) \cdot \left( \prod_{k=m}^{n} b_k \right) = \prod_{k=m}^{n} (a_k \cdot b_k) \]

**what’s not equal?**
Let \( a_k = k + 1 \) and \( b_k = k - 1 \) for all integers \( k \).

Write each of the following expressions as a single summation or product:

a. \[ \sum_{k=m}^{n} a_k + 2 \cdot \sum_{k=m}^{n} b_k \]

b. \[ \left( \prod_{k=m}^{n} a_k \right) \cdot \left( \prod_{k=m}^{n} b_k \right) \]
Consider
\[ \sum_{k=1}^{3} k^2 = 1^2 + 2^2 + 3^2 \]

and
\[ \sum_{i=1}^{3} i^2 = 1^2 + 2^2 + 3^2. \]

Hence
\[ \sum_{k=1}^{3} k^2 = \sum_{i=1}^{3} i^2. \]

k, i are symbols representing index of a summation.

They can be replaced by any other symbol as long as the replacement is made in each occurrence.
Transforming a Sum by a Change of Variable

summation: \( \sum_{k=0}^{6} \frac{1}{k+1} \)  
change of variable: \( j = k + 1 \)
i.e., new summation uses index \( j \)

1. lower and upper limits for \( j \)?

2. **formula** for new summation: replace each occurrence of \( k \) by an expression in \( j \) :

3. Finally put everything together:

\[
\sum_{k=0}^{6} \frac{1}{k+1} = \sum_{j=1}^{7} \frac{1}{j}.
\]
1. Transform following summation by making specified change of variable.

\[ \sum_{k=1}^{n+1} \left( \frac{k}{n+k} \right) \]

change of variable: \( j = k - 1 \)

2. Transform the summation obtained in part (1) by changing all \( j \)'s to \( k \)'s.
SECTION 5.6

Defining Sequences Recursively
Defining Sequences Recursively

Sometimes a sequence is defined using recursion.

- an equation, called a \textit{recurrence relation}, that defines each later term by reference to earlier terms
- together with one or more initial values for the sequence.

\textbf{Definition}

A \textit{recurrence relation} for a sequence \(a_0, a_1, a_2, \ldots\) is a formula that relates each term \(a_k\) to certain of its predecessors \(a_{k-1}, a_{k-2}, \ldots, a_{k-i}\), where \(i\) is an integer with \(k - i \geq 0\). The \textbf{initial conditions} for such a recurrence relation specify the values of \(a_0, a_1, a_2, \ldots, a_{i-1}\), if \(i\) is a fixed integer, or \(a_0, a_1, \ldots, a_m\), where \(m\) is an integer with \(m \geq 0\), if \(i\) depends on \(k\).
Define a sequence $c_0, c_1, c_2, \ldots$ recursively as follows: For all integers $k \geq 2$,

\begin{align*}
(1) \quad & c_k = c_{k-1} + kc_{k-2} + 1 \\
(2) \quad & c_0 = 1 \quad \text{and} \quad c_1 = 2
\end{align*}

recurrence relation

initial conditions.

Find $c_2, c_3,$ and $c_4$. 
Sequence of **Catalan numbers** (named after Belgian mathematician Eugène Catalan, 1814–1894), arises in different contexts in discrete mathematics. For each integer $n \geq 1$,

$$C_n = \frac{1}{n+1} \binom{2n}{n}.$$ 

1. Find $C_1, C_2,$ and $C_3$. 
Sequence of **Catalan numbers**

For each integer $n \geq 1$,

$$C_n = \frac{1}{n+1} \binom{2n}{n}.$$

Show this sequence satisfies recurrence relation

$$C_k = \frac{4k-2}{k+1} C_{k-1}$$

for all integers $k \geq 2$. 
Examples of Recursively Defined Sequences

Recursion is one of the central ideas of computer science.

It’s an approach to problem solving.
Examples of Recursively Defined Sequences

To solve a problem recursively means

1. to break original problem down into smaller subproblems each having same form as original problem

3. when the process is repeated many times, eventually the subproblems are small and easy to solve

5. solutions of the subproblems can be woven together to form a solution to the original problem.
The Tower of Hanoi

invented in 1883 by French mathematician, Édouard Lucas
• 3 poles, 8 disks of wood with holes in their centers, piled in order of decreasing size on one pole A

How to move all disks one by one from pole A to pole C, never placing a larger disk on top of a smaller one?
Think Recursively!

Suppose that you have found most efficient way possible to transfer a tower of $k - 1$ disks from one pole to another, obeying given restriction.

What is the most efficient way to transfer a tower of $k$ disks from one pole to another?
Solution: move $k$ disks $A$=>$C$

**Step 1:** move top $k - 1$ disks from $A$ to $B$. (If $k > 2$, this step will require a number of moves of individual disks among the three poles.)

*ignore existence of bottom disk*

**Step 2:** Move one disk from $A$ to $C$.

**Step 3:** move $k - 1$ disks from pole $B$ to pole $C$.

(Again, if $k > 2$, this step will require more than one move.)

*again ignore existence of bottom disk …*
To move bottom disk of a stack of $k$ disks from one pole to another, you **must** first transfer top $k - 1$ disks to a third pole to get them out of the way.

Transferring stack of $k$ disks from pole $A$ to pole $C$ **requires at least** two transfers of top $k - 1$ disks:

- one to transfer them off, to free the bottom disk so that it can be moved
- another to transfer them back on top of bottom disk after bottom disk has been moved to pole $C$. 

**Is this most efficient way?** cont’d
Solution

\[ m_n = \begin{cases} \text{the minimum number of moves needed to transfer} \\ \text{a tower of } n \text{ disks from one pole to another} \end{cases} \]

\[ m_n \] are independent of

- labeling of poles: it takes same minimum number of moves to transfer \( n \) disks from pole A to pole C, as to transfer \( n \) disks from pole A to pole B. ...

- independent of number of larger disks that may lie below top \( n \), provided these remain stationary while top \( n \) are moved.
How many moves?

It follows that

\[
\begin{align*}
\text{the minimum number of moves needed to transfer} & \quad \text{the minimum number of moves needed} \\
\text{a tower of } k \text{ disks} & \quad \text{to go from} \\
\text{from pole } A \text{ to} & \quad \text{position (a)} \quad \text{to position (b)} \\
\text{pole } C & \quad + \quad \text{The minimum number of moves needed} \\
& \quad \text{to go from} \\
& \quad \text{position (b)} \quad \text{to position (c)} \\
& \quad \text{to position (c)} \quad \text{to position (d)} \\
& \quad + \quad \text{the minimum number of moves needed} \\
& \quad \text{to go from} \\
& \quad \text{position (c)} \quad \text{to position (d)} \\
& \quad + \quad 5.6.1
\end{align*}
\]
How many moves?

It follows that

\[ m_k = m_{k-1} + 1 + m_{k-1} \]

\[ = 2m_{k-1} + 1 \]

for all integers \( k \geq 2 \).
Solution

Because just one move is needed to move one disk from one pole to another,

\[ m_1 = \left\lfloor \text{the minimum number of moves needed to move a tower of one disk from one pole to another} \right\rfloor = 1. \]

complete recursive specification of the sequence \( m_1, m_2, m_3, \ldots \) is as follows:

For all integers \( k \geq 2 \),

1. \( m_k = 2m_{k-1} + 1 \)  \hspace{1cm} \text{recurrence relation}

2. \( m_1 = 1 \)  \hspace{1cm} \text{initial conditions}
Here is a computation of the next five terms of the sequence:

(3) \( m_2 = 2m_1 + 1 = 2 \cdot 1 + 1 = 3 \) by (1) and (2)
(4) \( m_3 = 2m_2 + 1 = 2 \cdot 3 + 1 = 7 \) by (1) and (3)
(5) \( m_4 = 2m_3 + 1 = 2 \cdot 7 + 1 = 15 \) by (1) and (4)
(6) \( m_5 = 2m_4 + 1 = 2 \cdot 15 + 1 = 31 \) by (1) and (5)
(7) \( m_6 = 2m_5 + 1 = 2 \cdot 31 + 1 = 63 \) by (1) and (6)

Going back to the legend, suppose priests work rapidly and move one disk every second.

Then time from the beginning of creation to the end of the world would be \( m_{64} \) seconds.
Example 5 – Solution

We can compute $m_{64}$ on a calculator.
The approximate result is

$$1.844674 \times 10^{19} \text{ seconds} \approx 5.84542 \times 10^{11} \text{ years} \approx 584.5 \text{ billion years},$$

which is obtained by the estimate of

$$60 \cdot 60 \cdot 24 \cdot (365.25) = 31,557,600$$

seconds in a year (figuring 365.25 days in a year to take leap years into account). Surprisingly, this is close to some scientific estimates of the life of the universe!
Recursive Definitions of Sum and Product
Addition and multiplication are called *binary* operations because only two numbers can be added or multiplied at a time. Careful definitions of sums and products of more than two numbers use recursion.

---

**Definition**

Given numbers $a_1, a_2, \ldots, a_n$, where $n$ is a positive integer, the *summation from $i = 1$ to $n$ of the* $a_i$, denoted $\sum_{i=1}^{n} a_i$, is defined as follows:

\[
\sum_{i=1}^{1} a_i = a_1 \quad \text{and} \quad \sum_{i=1}^{n} a_i = \left( \sum_{i=1}^{n-1} a_i \right) + a_n, \quad \text{if } n > 1.
\]

The *product from $i = 1$ to $n$ of the* $a_i$, denoted $\prod_{i=1}^{n} a_i$, is defined by

\[
\prod_{i=1}^{1} a_i = a_1 \quad \text{and} \quad \prod_{i=1}^{n} a_i = \left( \prod_{i=1}^{n-1} a_i \right) \cdot a_n, \quad \text{if } n > 1.
\]
Recursive Definitions of Sum and Product

The effect of these definitions is to specify an order in which sums and products of more than two numbers are computed. For example,

\[
\sum_{i=1}^{4} a_i = \left( \sum_{i=1}^{3} a_i \right) + a_4 = \left( \left( \sum_{i=1}^{2} a_i \right) + a_3 \right) + a_4 = ((a_1 + a_2) + a_3) + a_4.
\]

The recursive definitions are used with mathematical induction to establish various properties of general finite sums and products.
SECTION 5.7

Solving Recurrence Relations by Iteration
It is often helpful to know an explicit formula for a sequence defined by a recurrence relation,

- if you need to compute terms with very large subscripts
- you need to examine general properties of the sequence.

Such an explicit formula is called a **solution** to the recurrence relation.
Method of Iteration

Iteration works as follows:

Given a sequence $a_0, a_1, a_2, \ldots$ defined by a recurrence relation and initial conditions,

you start from initial conditions and calculate successive terms of the sequence

until you see a pattern developing.

At that point you guess an explicit formula.
Finding an Explicit Formula

For all integers $k \geq 1$,

1. $a_k = a_{k-1} + 2$ \hspace{1cm} \text{recurrence relation}
2. $a_0 = 1$ \hspace{1cm} \text{initial condition.}

Use iteration to guess an explicit formula for the sequence.
The reason for using numerical expressions rather than numbers: as we are seeking a numerical pattern that underlies a general formula.

The secret of success is to leave most of the arithmetic undone.

Do eliminate parentheses as you go from one step to the next.
Example 1 – Solution

use shorthand notations for regrouping additions, subtractions, and multiplications of numbers that repeat.

\[ 5 \cdot 2 \quad \text{instead of} \quad 2 + 2 + 2 + 2 + 2 \]

and

\[ 2^5 \quad \text{instead of} \quad 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2. \]
A sequence like the one in Example 1, in which each term equals the previous term plus a fixed constant, is called an arithmetic sequence.

**Definition**

A sequence \( a_0, a_1, a_2, \ldots \) is called an arithmetic sequence if, and only if, there is a constant \( d \) such that

\[
 a_k = a_{k-1} + d \quad \text{for all integers } k \geq 1.
\]

It follows that,

\[
 a_n = a_0 + dn \quad \text{for all integers } n \geq 0.
\]
Let $r$ be a fixed nonzero constant, and suppose a sequence $a_0, a_1, a_2, \ldots$ is defined recursively as follows:

$$a_k = r a_{k-1} \quad \text{for all integers } k \geq 1,$$

$$a_0 = a.$$

Use iteration to guess an explicit formula for this sequence.
The Method of Iteration

An important property of a geometric sequence with constant multiplier greater than 1 is that its terms increase very rapidly in size as the subscripts get larger and larger.

Exponential growth!

For instance, first ten terms of a geometric sequence with a constant multiplier of 10 are

\[1, 10, 10^2, 10^3, 10^4, 10^5, 10^6, 10^7, 10^8, 10^9.\]

Thus, by its tenth term, the sequence already has the value \(10^9 = 1,000,000,000 = 1\) billion.
The Method of Iteration

following box indicates some quantities that are approximately equal to certain powers of 10.

\[
\begin{align*}
10^7 &\approx \text{number of seconds in a year} \\
10^9 &\approx \text{number of bytes of memory in a personal computer} \\
10^{11} &\approx \text{number of neurons in a human brain} \\
10^{17} &\approx \text{age of the universe in seconds (according to one theory)} \\
10^{31} &\approx \text{number of seconds to process all possible positions of a checkers game if moves are processed at a rate of 1 per billionth of a second} \\
10^{81} &\approx \text{number of atoms in the universe} \\
10^{111} &\approx \text{number of seconds to process all possible positions of a chess game if moves are processed at a rate of 1 per billionth of a second}
\end{align*}
\]
Explicit formulas obtained by iteration can often be simplified by using summation formulas, e.g.,

\[ 1 + r + r^2 + \cdots + r^n = \frac{r^{n+1} - 1}{r - 1} \quad \text{for all integers } n \geq 0. \]
And according to formula for the sum of the first $n$ integers,

$$1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2} \quad \text{for all integers } n \geq 1.$$
Example 5 – An Explicit Formula for the Tower of Hanoi Sequence

The Tower of Hanoi sequence $m_1, m_2, m_3, \ldots$ satisfies the recurrence relation

$$m_k = 2m_{k-1} + 1 \quad \text{for all integers } k \geq 2$$

and has the initial condition

$$m_1 = 1.$$ 

Use iteration to guess an explicit formula for this sequence, to simplify the answer.
Example 5 – Solution

By iteration

\[ m_1 = 1 \]

\[ m_2 = 2m_1 + 1 = 2 \cdot 1 + 1 = 2^1 + 1 \]

\[ m_3 = 2m_2 + 1 = 2(2 + 1) + 1 = 2^2 + 2 + 1 \]

\[ m_4 = 2m_3 + 1 = 2(2^2 + 2 + 1) + 1 = 2^3 + 2^2 + 2 + 1 \]

\[ m_5 = 2m_4 + 1 = 2(2^3 + 2^2 + 2 + 1) + 1 = 2^4 + 2^3 + 2^2 + 2 + 1 \]
These calculations show that each term up to $m_5$ is a sum of successive powers of 2, starting with $2^0 = 1$ and going up to $2^k$, where $k$ is 1 less than the subscript of the term.

The pattern would seem to continue to higher terms because each term is obtained from the preceding one by multiplying by 2 and adding 1; multiplying by 2 raises the exponent of each component of the sum by 1, and adding 1 adds back the 1 that was lost when the previous 1 was multiplied by 2.

For instance, for $n = 6$,

$$m_6 = 2m_5 + 1 = 2(2^4 + 2^3 + 2^2 + 2 + 1) + 1 = 2^5 + 2^4 + 2^3 + 2^2 + 2 + 1.$$
Thus it seems that, in general,

$$m_n = 2^{n-1} + 2^{n-2} + \cdots + 2^2 + 2 + 1.$$ 

By the formula for the sum of a geometric sequence (Theorem 5.2.3),

Theorem 5.2.3 Sum of a Geometric Sequence
For any real number $r$ except 1, and any integer $n \geq 0$,

$$\sum_{i=0}^{n} r^i = \frac{r^{n+1} - 1}{r - 1}.$$ 

$$2^{n-1} + 2^{n-2} + \cdots + 2^2 + 2 + 1 = \frac{2^n - 1}{2 - 1} = 2^n - 1.$$
Hence the explicit formula seems to be

\[ m_n = 2^n - 1 \quad \text{for all integers } n \geq 1. \]
Checking the Correctness of a Formula by Mathematical Induction

Come back to this later....
It is all too easy to make a mistake and come up with the wrong formula.

That is why it is important to confirm your calculations by checking the correctness of your formula.

The most common way to do this is to use mathematical induction.
In 1883 a French mathematician, Édouard Lucas, invented a puzzle that he called The Tower of Hanoi (La Tour D’Hanoï).

The puzzle consisted of eight disks of wood with holes in their centers, which were piled in order of decreasing size on one pole in a row of three.

Those who played the game were supposed to move all the disks one by one from one pole to another, never placing a larger disk on top of a smaller one.
The puzzle offered a prize of ten thousand francs (about $34,000 US today) to anyone who could move a tower of 64 disks by hand while following the rules of the game.

(See Figure 5.6.2) Assuming that you transferred the disks as efficiently as possible, how many moves would be required to win the prize?
The solution to this is as follows:

Let $m$ be the minimum number of moves needed to transfer a tower of $k$ disks from one pole to another. Then,

If $m_1, m_2, m_3, \ldots$ is the sequence defined by

$$m_k = 2m_{k-1} + 1 \quad \text{for all integers } k \geq 2, \text{ and}$$

$$m_1 = 1,$$

then $m_n = 2^n - 1$ for all integers $n \geq 1$.

Use mathematical induction to show that this formula is correct.
Example 7 – Solution

What does it mean to show the correctness of a formula for a recursively defined sequence? Given a sequence of numbers that satisfies a certain recurrence relation and initial condition, your job is to show that each term of the sequence satisfies the proposed explicit formula.

In this case, you need to prove the following statement:

If $m_1, m_2, m_3, \ldots$ is the sequence defined by

$$m_k = 2m_{k-1} + 1 \quad \text{for all integers } k \geq 2, \text{ and}$$
$$m_1 = 1,$$

then $m_n = 2^n - 1$ for all integers $n \geq 1$. 
Proof of Correctness:
Let \( m_1, m_2, m_3, \ldots \) be the sequence defined by specifying that \( m_1 = 1 \) and \( m_k = 2m_{k+1} + 1 \) for all integers \( k \geq 2 \), and let the property \( P(n) \) be the equation

\[
m_n = 2^n - 1 \quad \leftarrow P(n)
\]

We will use mathematical induction to prove that for all integers \( n \geq 1 \), \( P(n) \) is true.

Show that \( P(1) \) is true:
To establish \( P(1) \), we must show that

\[
m_1 = 2^1 - 1. \quad \leftarrow P(1)
\]
Example 7 – Solution

But the left-hand side of $P(1)$ is

$$m_1 = 1$$

by definition of $m_1, m_2, m_3, \ldots$,

and the right-hand side of $P(1)$ is

$$2^1 - 1 = 2 - 1 = 1.$$ 

Thus the two sides of $P(1)$ equal the same quantity, and hence $P(1)$ is true.
Show that for all integers $k \geq 1$, if $P(k)$ is true then $P(k + 1)$ is also true:

[Suppose that $P(k)$ is true for a particular but arbitrarily chosen integer $k \geq 1$. That is:]

Suppose that $k$ is any integer with $k \geq 1$ such that

$$m_k = 2^k - 1.$$  \hspace{1cm} \leftarrow P(k) \text{ inductive hypothesis}

[We must show that $P(k + 1)$ is true. That is:] We must show that

$$m_{k+1} = 2^{k+1} - 1.$$  \hspace{1cm} \leftarrow P(k + 1)$$
Example 7 – Solution

But the left-hand side of $P(k + 1)$ is

$$m_{k+1} = 2m_{(k+1)-1} + 1$$
$$= 2m_k + 1$$
$$= 2(2^k - 1) + 1$$
$$= 2^{k+1} - 2 + 1$$
$$= 2^{k+1} - 1$$

by definition of $m_1, m_2, m_3, \ldots$
by substitution from the inductive hypothesis
by the distributive law and the fact that $2 \cdot 2^k = 2^{k-1}$
by basic algebra

which equals the right-hand side of $P(k + 1)$. [Since the basis and inductive steps have been proved, it follows by mathematical induction that the given formula holds for all integers $n \geq 1$.]
Discovering That an Explicit Formula Is Incorrect
Discovering That an Explicit Formula Is Incorrect

The next example shows how the process of trying to verify a formula by mathematical induction may reveal a mistake.
Let $c_0, c_1, c_2, \ldots$ be the sequence defined as follows:

$$c_k = 2c_{k-1} + k \quad \text{for all integers } k \geq 1,$$

$$c_0 = 1.$$

Suppose your calculations suggest that $c_0, c_1, c_2, \ldots$ satisfies the following explicit formula:

$$c_n = 2^n + n \quad \text{for all integers } n \geq 0.$$

Is this formula correct?
Example 8 – Solution

Start to prove the statement by mathematical induction and see what develops.

The proposed formula passes the basis step of the inductive proof with no trouble, for on the one hand, \( c_0 = 1 \) by definition and on the other hand, \( 2^0 + 0 = 1 + 0 = 1 \) also.

In the inductive step, you suppose

\[
c_k = 2^k + k \quad \text{for some integer } k \geq 0
\]

and then you must show that

\[
c_{k+1} = 2^{k+1} + (k + 1).
\]
Example 8 – Solution

To do this, you start with $c_{k+1}$, substitute from the recurrence relation, and then use the inductive hypothesis as follows:

$$c_{k+1} = 2c_k + (k + 1) \quad \text{by the recurrence relation}$$

$$= 2(2^k + k) + (k + 1) \quad \text{by substitution from the inductive hypothesis}$$

$$= 2^{k+1} + 3k + 1 \quad \text{by basic algebra}$$

To finish the verification, therefore, you need to show that

$$2^{k+1} + 3k + 1 = 2^{k+1} + (k + 1).$$
Example 8 – Solution

Now this equation is equivalent to

\[ 2k = 0 \]

which is equivalent to

\[ k = 0 \]

by subtracting \( 2^{k+1} + k + 1 \) from both sides.

But this is false since \( k \) may be any nonnegative integer.

Observe that when \( k = 0 \), then \( k + 1 = 1 \), and

\[ c_1 = 2 \cdot 1 + 1 = 3 \quad \text{and} \quad 2^1 + 1 = 3. \]
Thus the formula gives the correct value for $c_1$. However, when $k = 1$, then $k + 1 = 2$, and

$$c_2 = 2 \cdot 3 + 2 = 8 \quad \text{whereas } 2^2 + 2 = 4 + 2 = 6.$$ 

So the formula does not give the correct value for $c_2$. Hence the sequence $c_0, c_1, c_2, \ldots$ does not satisfy the proposed formula.
Once you have found a proposed formula to be false, you should look back at your calculations to see where you made a mistake, correct it, and try again.