The principle of *mathematical induction*:

**If** statement $p(b)$ is true, and statement $p(n - 1) \Rightarrow p(n)$ is true for all $n > b$, **then** $p(n)$ is true for all integers $n \geq b$. 
We first prove the *base case*.

Then we assume $p(n - 1)$ is true—called the *inductive hypothesis* and abbreviated IH—and using this fact, we prove that $p(n)$ is true.
Fibonacci Numbers

The Fibonacci numbers

\[1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, \ldots\]

are generated by \( f(0) = 1, f(1) = 1, \) and

\[f(n) = f(n - 1) + f(n - 2).\]

We prove things about Fibonacci numbers using induction.
Fibonacci Counts

Fibonacci numbers count many things, such as:

number of ways of tiling a $2 \times n$ checkerboard with dominoes.

number of pairs of rabbits after $n$ months if each pair produces one new pair every month after their first month.

number of sequences of 0s and 1s of length $n$ without consecutive 1s.
Recurrence

A **recurrence** for a sequence expresses the next term as a function of the previous terms.
The recurrence \( g(n) = Ag(n - 1) + Bg(n - 2) \) has characteristic equation \( x^2 = Ax + B \).

The solution to the recurrence is a linear combination of \( r^n \) and \( s^n \) where \( r \) and \( s \) are the roots of the characteristic equation.
A **rooted tree** has **vertices** and **edges** with one vertex designated the **root** and all other vertices having a unique parent.

Rooted trees are normally drawn with the root at the top.

**A tree** is just like a rooted tree, except it does not have a special vertex.
Properties of Trees

(1) A tree is connected

(2) A tree contains no cycle

(3) Between any two vertices in a tree there is a unique path.

(4) If a tree has $n$ vertices, then it has $n-1$ edges.
A (simple) **graph** is a collection of vertices and edges such that each edge joins two vertices.

We do NOT allow **multiple edges** nor **self-loops**.
Graph Terminology

A **walk** is a sequence of vertices such that consecutive vertices are joined by an edge.

The **length** of a walk is the number of edges.

A **path** is a walk without repeated vertices.

A **cycle** is a walk without repeated vertices except that the first and last vertex are the same.

The terms **path** and **cycle** also refer to specific graphs.

A graph is **connected** if between every two vertices there is a walk.
The **degree** of a vertex is the number of edges coming out of it.

In a graph, the sum of the degrees is twice the number of edges.
Special Graphs

The **complete graph** $K_n$ on $n$ vertices has every pair of vertices joined by an edge.

The **complete bipartite graph** $K_{r,s}$ has $r$ vertices on one side, and $s$ vertices on the other. All edges go between the two sides.
An **Euler tour** is a walk that goes along every edge exactly once, and ends up where one started.

A connected graph has an Euler tour if and only if every vertex has even degree.
A Hamilton cycle is a cycle that visits every vertex exactly once.

The problem of determining whether a graph has a Hamilton cycle is hard in general.
A graph is **bipartite** if one can partition the vertices into two sets, such that each edge has an end in each set.

All trees are bipartite. A cycle is bipartite if and only if it has an even number of vertices.

A connected graph is bipartite if and only if every cycle has even length.
A **coloring** means assigning colors to each vertex such that no edge joins two vertices of the same color.

The **chromatic number** of a graph, denoted $\chi$, is the minimum number of colors needed for a coloring of the vertices.

The chromatic number of a graph is at most 1 more than the maximum degree.
Planar Graphs

A planar graph is one which has a drawing in the plane such that no pair of lines intersect.

For example, every tree is planar.
A **code** is a set of strings of the same length. Equivalently, a subset of the vertices of some hypercube.

The **Hamming distance** between two strings is the number of places they differ. The **weight** of a binary string is the number of 1’s.

The **distance** of the code is the minimum Hamming distance between strings.
A code is \textit{k-error-detecting} if and only if its distance is at least \(k + 1\).

A code is \textit{k-error-correcting} if and only if its distance is at least \(2k + 1\).
Check Bits

The **check-sum** of a string is the sum of all the bits, modulo 2.

In the **Hamming code**, there are $2^k - k - 1$ data bits and $k$ check bits $c_i$. Each $c_i$ is inserted in position $i$ and is then the check-bit for all other positions whose binary expansion has a 1 in position $i$. 
A linear code of length $n$ is a subspace of the vector space $\mathbb{Z}_2^n$ (that is, with arithmetic modulo 2).

If data $d$ is a binary vector, then the code-string $e$ that is sent is given by $e = dM$ where matrix $M$ is the generator matrix.
Reed–Muller Codes

Start with $R_1 = \{0, 1\}$. Code $R_{m+1}$ is obtained from $R_m$ by taking every string $w$ in $R_i$ and writing down both $ww$ and $w\overline{w}$, where $\overline{w}$ means string $w$ with all bits flipped.

The distance of $R_m$ is $2^{m-2}$. 