In Computer Science, recursion is a common strategy for solving problems.

- take a problem instance
- split it into subproblems ...
- ... until they are small

<u>ex</u> binary search

Problem: find an element in a sorted array.

$$A = (\underbrace{a_1, a_2, a_3}_{A_l}, \underbrace{\dots, \quad a_n}_{A_r})$$

$$A_{l_l} \quad A_{l_r} \qquad A_{r_l} \quad A_{r_r}$$

$$\vdots$$

base case: single element array.

Mathematical Induction is a proof technique that is analogous to recursion.

ex to prove that $1+2+3+\cdots+n = \frac{n(n+1)}{2}$,

$$P(n) = \begin{cases} 1+2+3+\cdots+n = \frac{n(n+1)}{2} & : T \\ \neq & : F \end{cases}$$

We prove that the formula holds for $n=0$ (base case) and that if it holds for $n \geq 1$, then it holds for $n+1$.

some specific

Let P be a predicate concerning ints ≥ 0. To give a proof by mathematical induction that $\forall n \geq 0 : P(n)$, we prove 2 things:

$\forall n \geq 0 \; P(n) \Rightarrow P(n+1)$

(1) Base case: $P(0)$

(2) Inductive case: $\forall n \geq 1$, prove that $P(n-1) \Rightarrow P(n)$

If we do (1) and (2), we've proved $\forall n \geq 0 : P(n)$.

why?

(5.1 in book)

ex Suppose we have proven $P(0)$ and $P(n-1) \Rightarrow P(n)$. These establish

$P(3)$.

Proof WTS $P(3)$.

Statement	reason
$P(0)$	assumption
$P(0) \Rightarrow P(1)$	plug in $n=1$ to $P(n-1) \Rightarrow P(n)$, assumption
$P(1)$	because $P(0) \Rightarrow P(1)$, and we have $P(0)$ (modus ponens)
$P(1) \Rightarrow P(2)$	plug in $n=2$ to $P(n-1) \Rightarrow P(n)$
$P(2)$	modus ponens
$P(2) \Rightarrow P(3)$	plug in $n=3$ to $P(n-1) \Rightarrow P(n)$
$P(3)$	modus ponens

Claim $\forall n \geq 0$, $\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$

$n \in \mathbb{Z}$

$$2^0 + 2^1 + 2^2 + \cdots + 2^n = 2^{n+1} - 1$$