

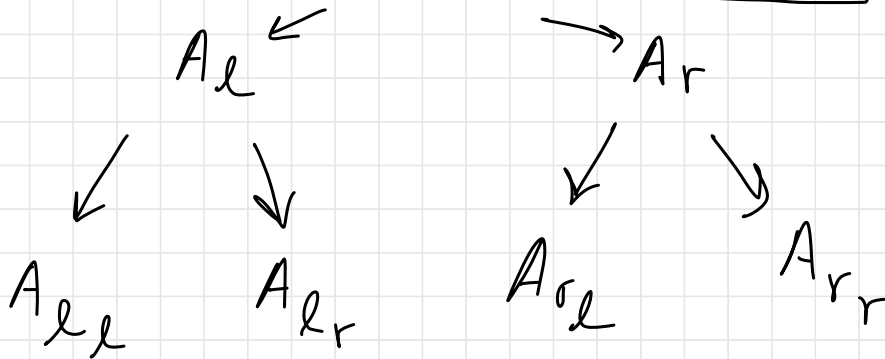
In Computer Science, recursion is a common strategy for solving problems.

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

ex binary search

Problem: find an element in a sorted array.

$$A = (\underbrace{a_1, a_2, a_3, \dots, a_n})$$



base case: single element array.

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

ex to prove that $1 + 2 + 3 + \dots + n$

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

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 $\text{ints} \geq 0$. 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

$$P(0)$$

$$P(0) \Rightarrow P(1)$$

$$P(1)$$

$$P(1) \Rightarrow P(2)$$

$$P(2)$$

$$P(2) \Rightarrow P(3)$$

$$P(3)$$

reason

assumption

plug in $n=1$ to
 $P(n-1) \Rightarrow P(n)$,
 assumption

because $P(0) \Rightarrow P(1)$,
 and we have $P(0)$,
 (modus ponens)

plug in $n=2$ to
 $P(n-1) \Rightarrow P(n)$

modus ponens

plug in $n=3$ to
 $P(n-1) \Rightarrow P(n)$

modus ponens

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

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

