

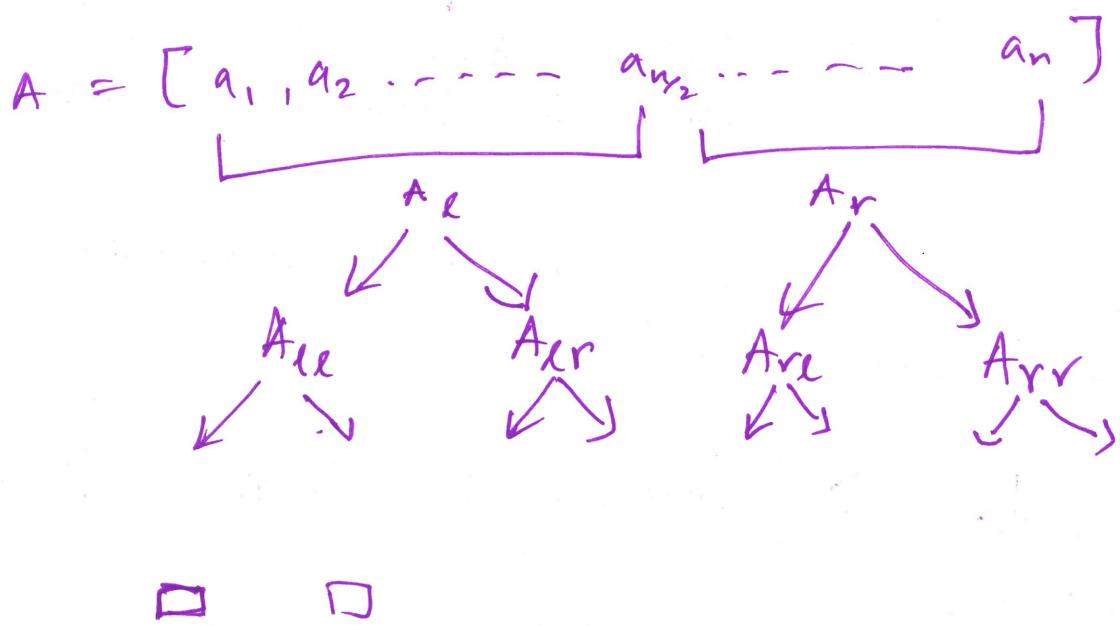
10/18/2024

Induction

- Recursion is a common strategy for problems in CS
 - Take a problem instance
 - split it into smaller subproblems
 - Once the subproblems are small enough that they tractable, you solve them.

Ex: Binary Search

- Given a sorted array, we want to find an element.



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

(This is only used to prove claims dealing with natural numbers)

$$\forall n \in \mathbb{N} : \sum_{i=0}^n i = \frac{n \cdot (n+1)}{2}$$

In induction, we do something like this step 1: We prove this claim for the smallest possible case.
(we prove for $n=0$)

Step 2: We assume that this claim holds for $n \geq 1$ and we show that the claim is true for $n+1$.

Let p be a predicate concerning integers greater than 0.

$p: \mathbb{N} \rightarrow \{\text{True}, \text{False}\}$, define $p(n)$ as follows.

$p(n)$ says $\sum_{i=0}^n i = \frac{n \cdot (n+1)}{2}$

$[\forall n \in \mathbb{N}: p(n)]$

I assume that I have already prove 2 things.

1. The base case (smallest case possible)

2. Inductive case ! $\forall n \geq 1 : P(n-1) \Rightarrow P(n)$

Let's see why these two steps helps us to prove [$\forall n : P(n)$]

Why?

Statement

$$P(0)$$

reason

We showed that the base case is true.

using inductive case and plugging ~~$n=1$~~ .

"Modus ponens"
Since we already know $P(0) \Rightarrow P(1)$, therefore if $P(0)$ is true, $P(1)$ has to be true.

"this way that affirms by affirming."

P	q	$P \Rightarrow q$
0	0	1
0	1	1
1	0	0
1	1	1

If $P(0) \Rightarrow P(1) \wedge P(0)$, then $P(1)$ has to be true

(this is the only case to consider if $[P(0) \Rightarrow P(1)]$ is true)

→ "continued on Monday".

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

using the inductive case
& plugging $n=2$

$P(2)$

using modus ponens.

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

using inductive case
& plugging $n=3$
modus ponens.

$P(3)$

:

:

:

:

:

:

$P(n)$

modus ponens.

$\underbrace{P \Rightarrow q}_{\text{True}}$

P is false

what can you say
about q ?

$$\forall n \geq 0: \sum_{i=0}^n i = \frac{n \cdot (n+1)}{2}$$

Step 01: Define the predicate $P(n)$

$P: \mathbb{N} \rightarrow \{\text{T}, \text{F}\}$, $P(n)$ is defined as follows:

$$P(n) = \begin{cases} \text{T}, & \text{if } \sum_{i=0}^n i = \frac{n \cdot (n+1)}{2} \\ \text{F}, & \text{otherwise} \end{cases}$$

Step 02: state the variable that you are performing induction

- we perform induction over $n \in \mathbb{N}$

we want to show: $[\forall n \in \mathbb{N}: P(n)]$

Step 03: State the base case

$n=0$ is the base case.

Step 04: Prove the base case

$$\text{WTS: } \sum_{i=0}^0 i = \frac{0 \times (0+1)}{2}$$

$\underbrace{\quad}_{\text{LHS}}$ $\underbrace{\frac{0}{2}}_{\text{RHS}}$

$$\text{LHS} = \sum_{i=0}^0 i = 0$$

$$\text{RHS} = \frac{0 \times 1}{2} = 0$$

$$\text{LHS} = \text{RHS}$$

\therefore base case is proven.

Step 05: State the inductive case

$$\forall n \geq 1 : P(n-1) \Rightarrow P(n)$$

This
is
okay

$$\boxed{\forall n \geq 0 : P(n) \Rightarrow P(n+1)}$$

$$\left| \begin{array}{l} \forall n \geq 1 : \left(\sum_{i=0}^{n-1} i = \frac{(n-1)n}{2} \right) \Rightarrow \sum_{i=0}^n i = \frac{n(n+1)}{2} \\ \text{LHS} \quad \quad \quad \text{RHS} \end{array} \right.$$

Step 06: Prove the inductive case

$$\text{Let } n \geq 1$$

Assume $P(n-1)$ is true ← Inductive assumption.

$$\begin{aligned} \text{LHS} &= \sum_{i=0}^n i && (\text{def of summation}) \\ &= \sum_{i=0}^{n-1} i + n && (\text{def of summation}) \end{aligned}$$

$$= \frac{(n-1) \cdot n}{2} + n \quad \text{substituting inductive hypothesis.}$$

$$\text{LHS} = \frac{n(n-1) + 2n}{2} = n \frac{(n-1+2)}{2} = \frac{n \cdot (n+1)}{2} \quad \text{by algebra}$$

$$\text{LHS} = \text{RHS}$$

We have proven $P(n)$ is true, when $P(n-1)$ is true.

Step 07: we have shown that $P(0)$ (base case) is true, and $\forall n \geq 1 : P(n-1) \Rightarrow P(n)$.

∴ using the principle of mathematical induction $[\forall n \geq 0 : P(n)]$ is true //

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

↙
Sum of first ($n+1$) terms of powers of 2

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

$\underbrace{\hspace{10em}}$
 $(n+1)$ terms

Step 01: Define the predicate $P(n)$

$P: \mathbb{N} \rightarrow \{T, F\}$ $P(n)$ is defined as follows:

$$P(n) = \begin{cases} T & , \text{if } \sum_{i=0}^n 2^i = 2^{n+1} - 1 \\ F & , \text{otherwise} \end{cases}$$

Step 02: Define the variable that you perform induction.

Variable $n \in \mathbb{N}$

→ $[\forall n \in \mathbb{N}: P(n)]$

Step 03: ~~Prove~~ state the base case

$$n=0$$

Step 04: Prove the base case.

WTS: $\sum_{i=0}^0 2^i = 2^{0+1} - 1$

$\underbrace{\hspace{2em}}_{LHS} \quad \underbrace{\hspace{2em}}_{RHS}$

$$LHS = \sum_{i=0}^0 2^i = 2^0 = 1$$

$$RHS = 2^1 - 1 = 1$$

$$LHS = RHS //$$