10/18/2024

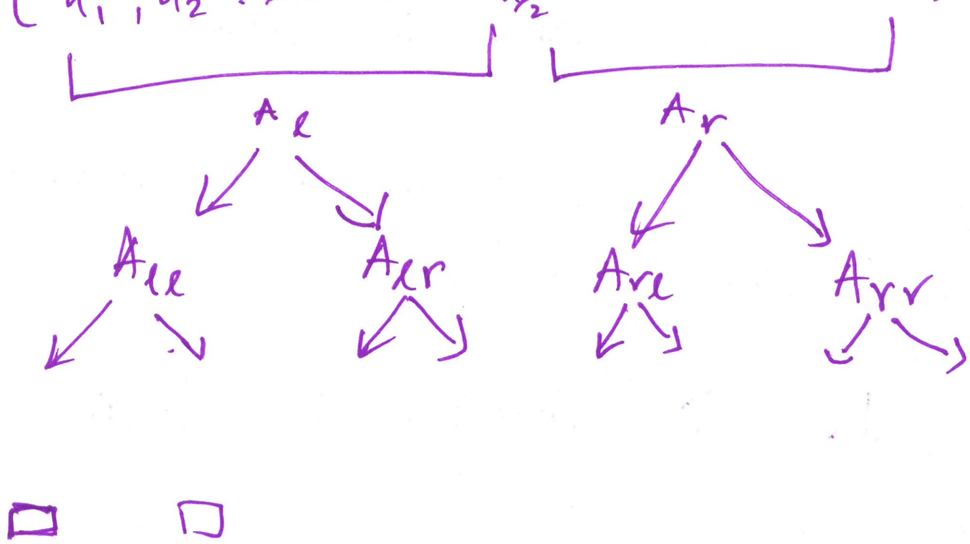
Induction

- 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.

$$A = [a_1, a_2 \cdots\cdots a_{n/2} \cdots\cdots a_n]$$



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

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

$$\forall n \in \mathbb{N} : \underbrace{0 + 1 + 2 + 3 + \dots + n}_{\sum_{i=0}^{n} i} = \frac{n \cdot (n+1)}{2}$$

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.

$P : \mathbb{N} \longrightarrow \{True, False\}$, 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 $[\forall n \in \mathbb{N} : P(n)]$

Why?

Statement	reason
$P(0)$	we showed that the base case is true.
$P(0) \Rightarrow P(1)$	using inductive case and plugging ~~n=1~~ $n=1$
$P(1)$ is true	"Modus ponens" Since we already know $P(0) \Rightarrow P(1)$, therefore if $P(0)$ is true, $P(1)$ has to be true. "the way that affirms by affirming"

p	q	$p \Rightarrow q$
0	0	1
0	1	1
1	0	0
1	1	[illegible]

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$]$)