Recursion

Main idea: just like mathematical induction.

Normal proof:

$H \Rightarrow A \Rightarrow B \Rightarrow C$ ✓

(SAS) algebra ...

$H \Rightarrow C$.

(if H, then C).

Induction: new proof technique.

Often applied to statements about natural numbers $(1, 2, 3, \ldots)$

Goal: prove property $S(n)$ is true for all $n \in \mathbb{N}$

$$\left(\text{E.g., } S(n) = \sum_{i=1}^{n} i = \frac{n(n+1)}{2}\right)$$

New technique: prove a parameterized implication: $S(n) \Rightarrow S(n+1)$. (*)

Then prove explicitly that say $S(1)$ is true.

Now for any $n \in \mathbb{N}$, $S(n)$ must also be true:

$$S(1) \overset{(*)}{\Rightarrow} S(2) \overset{(*)}{\Rightarrow} S(3) \Rightarrow \cdots \overset{(*)}{\Rightarrow} S(n) \checkmark$$

↑ explicit (aka "base case")

Example: Show $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$.

Explicit base case: $n=1$:

$$\sum_{i=1}^{1} i = 1 = \frac{1(1+1)}{2} \checkmark$$

Now assume the result for n. I.e., $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$. Show $\sum_{i=1}^{n+1} i = \frac{(n+1)(n+2)}{2}$.

"inductive hypothesis"

$$\sum_{i=1}^{n+1} i = (n+1) + \sum_{i=1}^{n} i = (n+1) + \frac{n(n+1)}{2} \quad (\text{ind. hyp.})$$

$$= \frac{2(n+1) + n(n+1)}{2}$$

$$= \frac{n^2 + 3n + 2}{2} = \frac{(n+1)(n+2)}{2} \checkmark$$

Connection to programming: we're going to write functions that call themselves.

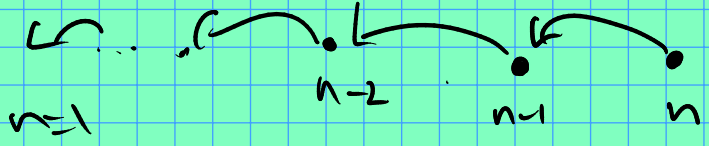
What's $S(n)$?

$S(n) \equiv$ my function "works" on inputs of size n.

Example: a function to compute $n! = 1 \cdot 2 \cdot 3 \cdots n$.

```
int fac(int n)
{  if (n<2) return 1; // base case.
   // now assume fac. works on any
   // input ≤ n. (In particular, n-1)
   return n * fac(n-1);
}
```

$n \cdot (n-1)!$



Let's trace a call to fac(4):