

We can use Big-O proofs to reason about complexity (of our programs)
(usually time)
↑
rates of growth

We can use mathematical induction to reason about correctness

Correct

Function isSorted(A, n):

For i From 0 to n-2:

 If $A[i] > A[i+1]$:

 Return False

 End If

End For

 Return True

End Function

Wrong

Function isSorted(A, n):

 If $n > 1$ And $A[0] > A[1]$:

 Return False

 End If

 Return True

End Function

Mathematical Induction

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

$\underbrace{\hspace{10em}}_{n+1}$
 $\underbrace{\hspace{10em}}_{n+1}$
.....

$$P(n) \equiv \sum_{i=1}^n i = \frac{n(n+1)}{2}$$

"proposition" - statement either true or false

"proposition function" - function giving back a proposition
not true for all n

\rightarrow
 $P(n) \equiv$ "it is the year n "

1. State P
2. Prove $P(0)$ (or $P(1)$ or other base case)
3. Prove, given $P(k)$ true, that $P(k+1)$ is true (without knowing k)
4. Invoke principle of mathematical induction

$P(0)$ by step 2

$P(1)$ by step 3

$P(2)$ by step 3

⋮

$$1. P(n) \equiv \sum_{i=1}^n i = \frac{n(n+1)}{2}$$

2. Base case. We want to show $P(1)$ is true.

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

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



3. Inductive step. WTS $P(k)$ implies $P(k+1)$.

$$\text{WTS if } \sum_{i=1}^k i = \frac{k(k+1)}{2} \text{ then } \sum_{i=1}^{k+1} i = \frac{(k+1)(k+2)}{2}$$

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

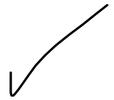
$$\left(\sum_{i=1}^k i \right) + k+1 = \frac{k(k+1)}{2} + k+1$$

$1+2+\dots+k-1+k$

$$\sum_{i=1}^{k+1} i$$

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

$$1+2+\dots+(k-1)+k+(k+1) = \frac{k(k+1) + 2(k+1)}{2} = \frac{(k+1)(k+2)}{2}$$



4. Therefore, by principle of mathematical induction, $\sum_{i=1}^n i = \frac{n(n+1)}{2}$ for all integers $n \geq 1$.

1. State P
2. Prove base case (e.g. $P(1)$)
3. Prove inductive step ($P(k) \Rightarrow P(k+1)$)
4. Wrap up

$$1. P(n) \equiv \sum_{i=0}^n 2^i = 2^{n+1} - 1$$

$$2. \text{WTS } P(0). \quad \text{WTS } \sum_{i=0}^0 2^i = 2^{0+1} - 1.$$

$$1 = 2 - 1 \quad \checkmark$$

$$3. \text{WTS } \forall k \geq 0 \in \mathbb{Z}. P(k) \Rightarrow P(k+1).$$

$$\text{WTS } \text{If } \sum_{i=0}^k 2^i = 2^{k+1} - 1 \text{ then } \sum_{i=0}^{k+1} 2^i = 2^{k+2} - 1.$$

$$\left(\sum_{i=0}^k 2^i \right) + 2^{k+1} = 2^{k+1} + 2^{k+1} - 1$$

$$\sum_{i=0}^{k+1} 2^i = 2 \cdot 2^{k+1} - 1 = 2^{k+1+1} - 1 = 2^{k+2} - 1$$

$$4. \text{By PMI, for all } n \geq 0 \in \mathbb{Z}, \sum_{i=0}^n 2^i = 2^{n+1} - 1$$

for all $n \geq 0$ that is an integer

Function $\text{summate}(n)$:

If $n == 1$:

Return n

Else:

Return $\text{summate}(n-1) + n$

End If

End Function

$P(n) \equiv$ When summate is called with a number n , it returns $\sum_{i=1}^n i$.

$P(1) \equiv$ When summate is called with a number 1, it returns 1.

When summate is called with 1, the IF statement evaluates its condition to true. So we run "Return n ", which returns 1.

$P(k) \Rightarrow P(k+1)$

inductive hypothesis

When summate is called w/ $k+1$ (for positive k), $n = k+1$. So the If condition is false. So we run the Else block. By inductive hypothesis, $\text{summate}(n-1)$ will return $\sum_{i=1}^k i$. So we return $\left(\sum_{i=1}^k i\right) + n = \sum_{i=1}^{k+1} i$.