

Problem 1

Write a recursive function that given an input n sums all non-negative integers up to n

$$\text{sum}(0) \rightarrow 0$$

$$\text{sum}(1) \rightarrow 1$$

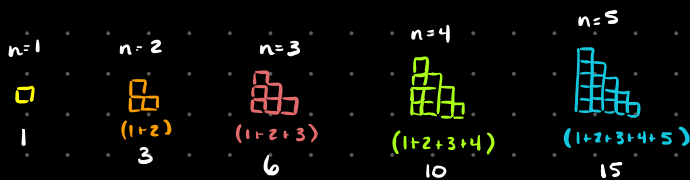
$$\text{sum}(4) \rightarrow (1+2+3+4) \rightarrow 10$$

$$\text{sum}(n) \rightarrow (1+2+3+\dots+n)$$

1. What's the simplest possible input?

$$\text{sum}(0) \rightarrow 0 \quad (\text{Base Case!})$$

2. Play around w/ examples and visualize



3. Relate hard cases to simpler cases!

What is the relationship between all these?

$$\text{sum}(4) + \text{sum}(5)$$

$$10 + 5(n) = 15$$

↳ if we know answer to $\text{sum}(4)$, just add 5 to it to get $\text{sum}(5)$

4. Generalize pattern

$$\begin{array}{c} n=k \\ \text{Visual representation of sum}(k) \end{array} = \begin{array}{c} n=k-1 \\ \text{Visual representation of sum}(k-1) \end{array} + \frac{\text{Visual representation of } k}{k}$$

sum(k)

sum(k-1)

5. Write code by combining recursive pattern w/ base case.

$$\text{sum}(n) = \begin{cases} 0 & \text{if } n = 0 \\ \text{sum}(n-1) + n & \end{cases}$$



```
def sum(n):  
    if n == 0:  
        return 0  
    else:  
        return n + sum(n-1)
```

Alton's Notes Covers Rest of Video!