

Q: What is a recurrence relation for the runtime of this algorithm?

Thing(n)

if $n=1$ or $n=2$: return n

for $i=1$ to n^2

Print: "meh"

return Thing($n-1$) + Thing($n-2$)

A) $T(1) = O(1)$; $T(n) = T(n-1) * T(n-2) + O(n^2)$

B) $T(1) = O(n)$; $T(n) = T(n-1) * T(n-2) + O(n^2)$

C) $T(1) = O(1)$; $T(n) = T(n-1) + T(n-2) + O(n^2)$

D) $T(1) = O(1)$; $T(2) = O(1)$; $T(n) = T(n-1) + T(n-2) + O(n^2)$

Q: What is a recurrence relation for the runtime of this algorithm?

Thing(n)

if n=1 or n=2: return n ← O(1)

for i=1 to n²] ← O(n²)

Print: "meh"

multiplication,
return
↓

return Thing(n-1) * Thing(n-2) ← T(n-1) + T(n-2) + O(1)

A) $T(1) = O(1); T(n) = T(n-1) * T(n-2) + O(n^2)$

B) $T(1) = O(n); T(n) = T(n-1) * T(n-2) + O(n^2)$

C) $T(1) = O(1); T(n) = T(n-1) + T(n-2) + O(n^2)$

D) $T(1) = O(1); T(2) = O(1); T(n) = T(n-1) + T(n-2) + O(n^2)$

need 2 base cases because otherwise fall off the bottom of ladder

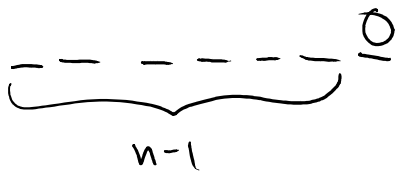
$T(1) = O(1)$

$T(2) = T(1) + T(0) + O(2^2)$

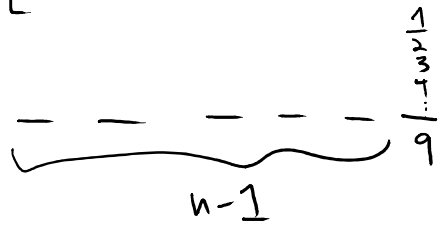
⇕
?

$K(n)$ = set of n -digit #'s with an even # of 0's.

$$K(n) = \left[\begin{array}{l} \text{even \# of 0's} \\ \text{end with 0} \end{array} \right] + \left[\begin{array}{l} \text{even \# of 0's} \\ \text{end with not 0} \end{array} \right]$$



Need this part to have odd # of 0's.



Need this part to have even # of 0's

$$\Rightarrow K(n-1) \rightarrow 9K(n-1)$$

$$\left[\begin{array}{l} \# \text{ n-1 digit with} \\ \text{odd \# of 0's} \end{array} \right] + \left[\begin{array}{l} \# \text{ n-1 digit} \\ \text{with even \#} \\ \text{of 0's} \end{array} \right] = \text{All} = 10^{n-1}$$



$$= 10^{n-1} - K(n-1)$$

$$\uparrow \\ K(n-1)$$

10 options for each digit

$$K(n) = 10^{n-1} - K(n-1) + 9K(n-1) = 10^{n-1} + 8K(n-1)$$

Let $T(n, k) = \#$ of strings in $\{0, 1, 2\}^n$ whose digits sum to k :

Options

$$\begin{array}{l}
 A: _ _ _ _ _ 0 \quad \leftarrow T(n-1, k) \\
 B: _ _ _ _ _ 1 \quad \leftarrow T(n-1, k-1) \quad \leftarrow \text{rest needs to sum to } k-1 \\
 C: _ _ _ _ _ 2 \quad \leftarrow T(n-1, k-2) \quad \leftarrow \text{rest needs to sum to } k-2 \\
 \quad \underbrace{\hspace{10em}}_{n-1}
 \end{array}$$

$$T(n, k) = T(n-1, k) + T(n-1, k-1) + T(n-1, k-2)$$

Base cases

$$T(n, 0) = 1 \quad (\text{all } 0\text{'s string})$$

$$T(n, \text{negative}) = 0 \quad (\text{never will sum to negative } \#)$$