PW SKILLS

Recursion

↳ When any function calls itself

Method name

$$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$$

fact(n):

Base condition

if $n \leq 1$

↳ 1

↳ O(1)

$n = 0$

$1! = 1, \; 0! = 1$

else: $n > 1$

return $n * \text{fact}(n-1)$

Recursion

Recursive Tree

~~fact~~(5) → 120

$n = 5$

function calls

↳ $5 * \cancel{\text{fact}(4)}$ → 24

↳ $4 * \cancel{\text{fact}(3)}$ → 6

↳ $3 * \text{fact}(2)$ → 6, 2

↓

$2 * \text{fact}(1)$ → 1

Stack Data Structure

~~fact(1)~~
~~fact(2)~~
~~fact(3)~~
~~fact(4)~~
~~fact(5)~~

O(n)

Note: for every recursive code, there exists a recurrence relation

↳ 1) Substitution Method

2) Recursive tree Method

3) Master's Theorem

Substitution Method

Example 1

$$\boxed{T(n) = 2T\left(\frac{n}{2}\right) + 4n} \quad \text{— (1)}$$

$$T\left(n/2\right) = 2T\left(\frac{n}{2^2}\right) + 4\left(\frac{n}{2}\right)$$

$$T(n) = 2\left(2T\left(\frac{n}{2^2}\right) + 4\left(\frac{n}{2}\right)\right) + 4n$$

$$T(n) = 2^2\,T\left(\frac{n}{2^2}\right) + 4n + 4n$$

$$\boxed{T(n) = 2^2\,\boxed{T\left(\frac{n}{2^2}\right)} + 2 * 4n} \quad \text{— (2)}$$

$$T\left(\frac{n}{2^2}\right) = 2T\left(\frac{n}{2^3}\right) + 4\left(\frac{n}{2^2}\right)$$

$$T(n) = 2^2\left(2\,T\left(\frac{n}{2^3}\right) + 4\left(\frac{n}{2^2}\right)\right) + 2*(4n)$$

$$T(n) = 2^3\,T\left(\frac{n}{2^3}\right) + 4n + 2*(4n)$$

$$\boxed{T(n) = 2^3\,T\left(\frac{n}{2^3}\right) + 3*(4n)} \quad \text{--- } (3)$$

$$\Big\downarrow \text{ k times}$$

$$T(1) = 1$$

$$\boxed{T(n) = 2^k\,\boxed{T\left(\frac{n}{2^k}\right)} + k*(4n)}$$

$$\frac{n}{2^k} = 1$$

$$\boxed{k = \log_2 n}$$

$$n = 2^k$$

$$\log_2 n = k\,\cancel{\log_2 2}$$

$$T(n) = 2^{\log_2 n} T\left(\frac{n}{2^{\log_2 n}}\right) + (4n) * \log_2 n$$

$$\boxed{a^{\log_2 b} = b^{\log_2 a}}$$

$$T(n) = n^{\log_2 \cancel{2}} T\left(\frac{\cancel{n}}{\cancel{n^{\log_2 2}}}\right) + (4n) * \log_2 n$$

c

$$T(n) = \underline{n} + (4n) * \underline{\log_2 n}$$

Time complexity $\Rightarrow$ $\boxed{T(n) = O(n \log_2 n)}$

Loop $\longrightarrow$ Iterative approach

Recursion $\longrightarrow$ Recursive approach

↳ Substitution Method

Example 2

1, 7, 31, 127, 511

1 2 3 4 5

$T(1) = 1$

$$\begin{aligned}
T(2) &= 4 * T(n-1) + 3 \\
&= 4 * T(1) + 3 \\
&= 4 * 1 + 3 \\
&= 7
\end{aligned}$$

$$\begin{aligned}
T(3) &= 4 * T(n-1) + 3 \\
&= 4 * T(2) + 3 \\
&= 4 * 7 + 3 \\
&= 31
\end{aligned}$$

$$\begin{aligned}
T(4) &= 4 * T(n-1) + 3 \\
&= 4 * T(3) + 3 \\
&= 4 * 31 + 3 \\
&= 124 + 3 \\
&= 127
\end{aligned}$$

$$\begin{aligned}
T(5) &= 4 * T(n-1) + 3 \\
&= 4 * T(4) + 3 \\
&= 4 * 127 + 3 \\
&= 508 + 3 \\
&= 511
\end{aligned}$$

$$T(n) = \begin{cases} 1 & n=1 \\ 4 * T(n-1) + 3 & n \geq 1 \end{cases}$$

↳ Recurrence Relation