PW SKILLS

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

$T(n)$

```
factorialfind(int n){
    if(n==0 || n==1){
        return 1;
    }
    else{
        int result = n * factorialfind(n-1);
    }
}
```

Base case condition

Recursive code

Recursive call

$O(n)$

Linear Time Complexity

$n \geq 0$

Recurrence Relation

$$T(n) = \begin{cases} 1 & n \leq 1 \\ T(n-1) + 1 & n > 1 \end{cases}$$

Substitution

$$T(n) = T(n-1) + 1 \quad \text{---} \quad (1)$$

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

$$T(n-2) = T(n-3) + 1$$

$$T(n) = T(n-2) + 2 \quad \text{---} \quad (2)$$

$$T(n) = T(n-3) + 3 \quad \text{---} \quad (3)$$

k times

$n - K = 1$

$n - 1 = K$

$T(n) = T(n-K) + K$ ——— k times

$T(n) = T(n-(n-1)) + (n-1)$

$T(n) = T(n-n+1) + (n-1)$

$$\boxed{T(n) = O(n)}$$

Recursive Tree

factorialfind(5) ← 120

5 * factorialfind(4) ← 24 = 120

4 * factorialfind(3) ← 6 = 24

3 * factorial find(2) ← 2 = 6

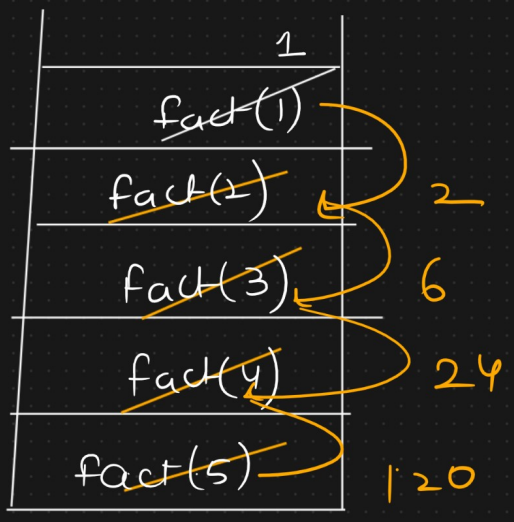
2 * factorialfind(1) ← 2

1

Data Structure

Stack ⟶ Recursive code

(LIFO) Last In First Out



Empty Stack