Recursive

$5 * 2 = 5 + 5 = 10$

$a_1 + a_2 + a_3 + \cdots a_b$

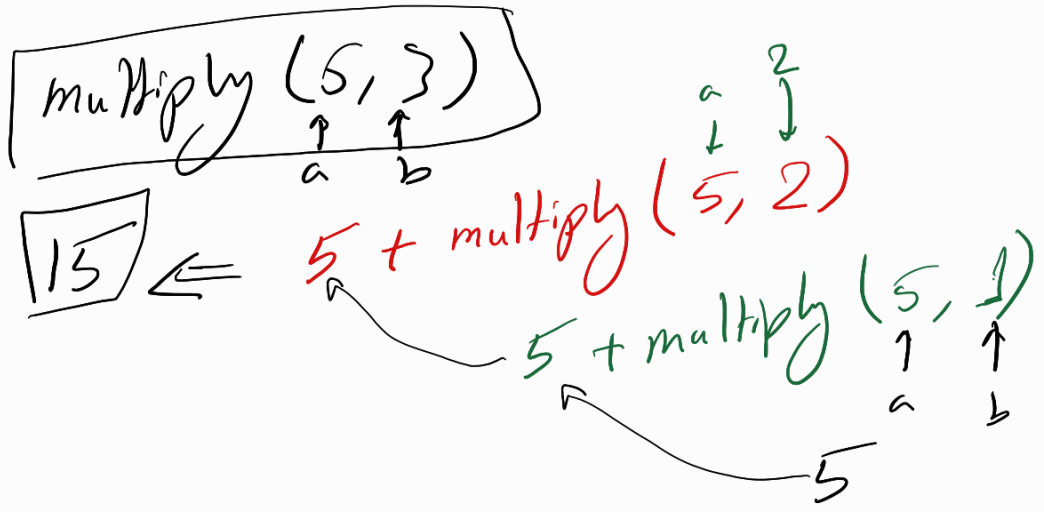
multiply (a, b)

$$\begin{cases} 0, & b = 0 \\ a, & b = 1 \\ a + \text{multiply}(a, b-1), & b > 1 \end{cases}$$

→ base case

→ recursive case

```
def multiply(a, b):
    if b == 0:
        return 0
    elif b == 1:
        return a
    else:
        return a + multiply(a, b-1)
```



Factorial

$n! = n(n-1)(n-2)(n-3) \cdots (1)$

$0! = 1$

$$f(n) = \begin{cases} 1, & n = 0, n-1 \\ n * f(n-1), & n > 1 \end{cases}$$

$n = 0, n-1 \Rightarrow$ Base

$n > 1 \Rightarrow$ recursive

```
def fact(n):
    if n == 0 or n == 1:
        ⇒ return 1
    return n * fact(n-1)
```

fact(4)

24 <= 4 * fact(3)

6 <= 3 * fact(2)

2 <= 2 * fact(1)

1

Fibonacci fib(4)

fib(0) fib(1) fib(2) fib(3) fib(4) fib(5)

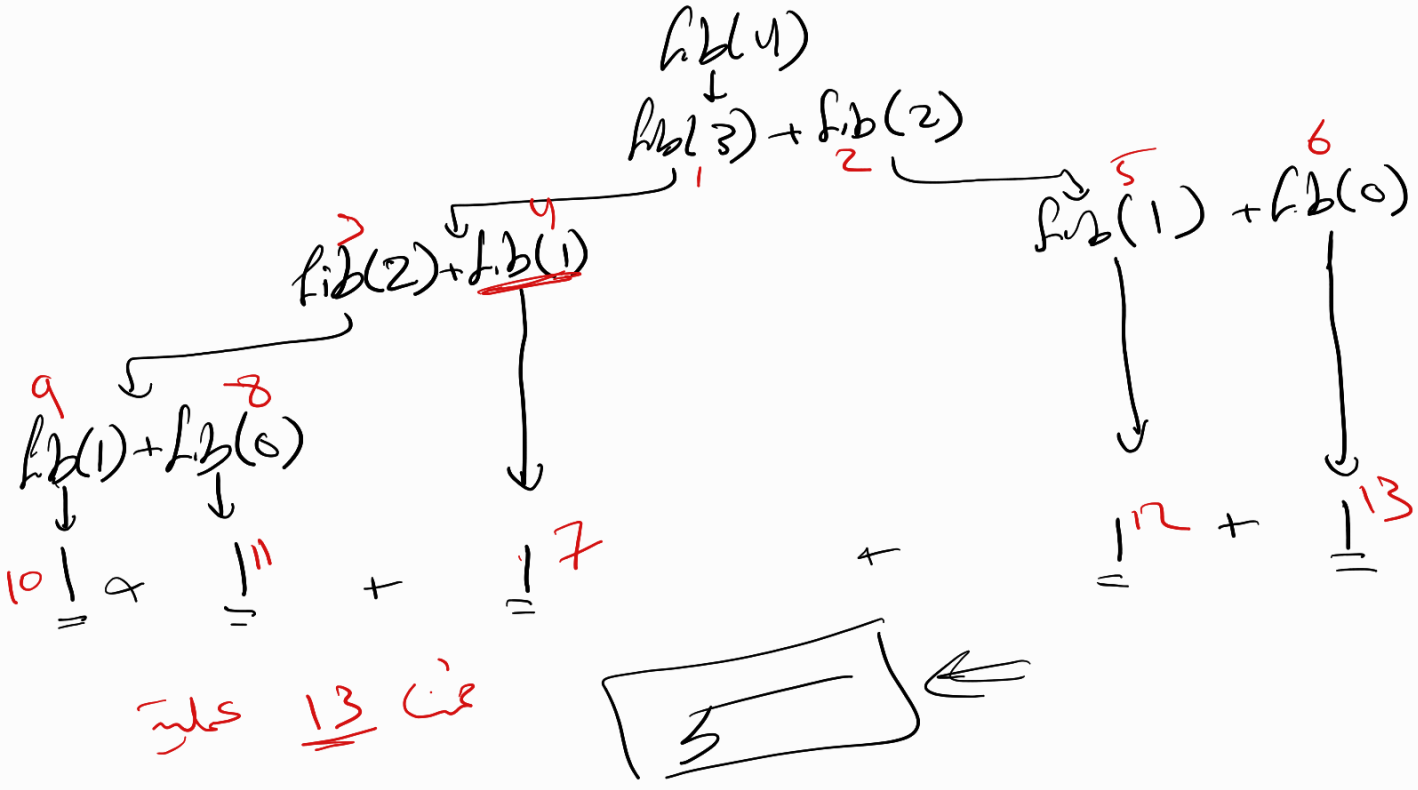
1 1 2 3 5 8

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

$$fib(n) = \begin{cases} 1, & n=0,\ n=1 \\ fib(n-1) + fib(n-2), & n>1 \end{cases}$$

```
def fib(n):
    if n == 0 or n == 1:
        return 1
```

return fib(n-1) + fib(n-2)



0	1	2	3	4	5
0	1	1	2	3	5
		2	3	5	8

```
def fib_loop(n):
    t1 = 0
    t2 = 1
    for _ in range(n):
        sum = t1 + t2
        t1 = t2
        t2 = sum
    return t2
```

0,1,2,3

fib_loop(4)

t_1	t_2	sum	_
0	1	1	0 ✓
1	1	2	1 ✓
1	2	3	2 ✓
2	3	5	3 ✓
3	5		4

عدد العمليات ← 4

$$S(5) = 1+2+3+4+5$$

$$S(n) = 1+2+ \cdots + n-1 + n$$

①

```
s=0
for i in range(1, n+1):
    s += i
```

O(n) ✓

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

```
def sum(n):
    return n*(n+1)/2
```

O(3) ⇒ O(c) → O(1)

③

```
s=0
for i in range(1, n+1):
    for j in range(1, i+1):
        s += 1
```

[1, 2)

→ 4

10 ← 1+2+3+4

i	j	count
1	1	✓ 1
2	1,2	✓ 2
3	1,2,3	✓ 3
4	1,2,3,4	✓ 4

1+2+3+4

s: ~~1~~, ~~3~~, ~~6~~, 10

$$\frac{y(y+1)}{2} = 10$$

$$\frac{n(n+1)}{2} \longrightarrow \frac{n^2+n}{2}$$

$$\Rightarrow O\left(\frac{n^2+n}{2}\right)$$

$$\hookrightarrow O(n^2)$$

→ for i in range(5) ↙ 1000000

→ for j in range(4) 1000000

a += 1 ⟸

25 ↙

n^2 ↙

$O(n^2)$

nested loops

---

for i in range(5)

a+= i ⟶ n

for i in range(5)

a += i ⟶ n

$n + n = 2n \Longrightarrow O(n)$

Consecutive stmts

```
s = k = 0
for i in range(n):
    if i < 3:
        st = i
    if i >= 3:
        k += i
```

n * 3 = 3n ⇒ O(3n) → O(n)

Computer is computed using CPU clock

O(n) → CPU unit

O(n²) → CPU units