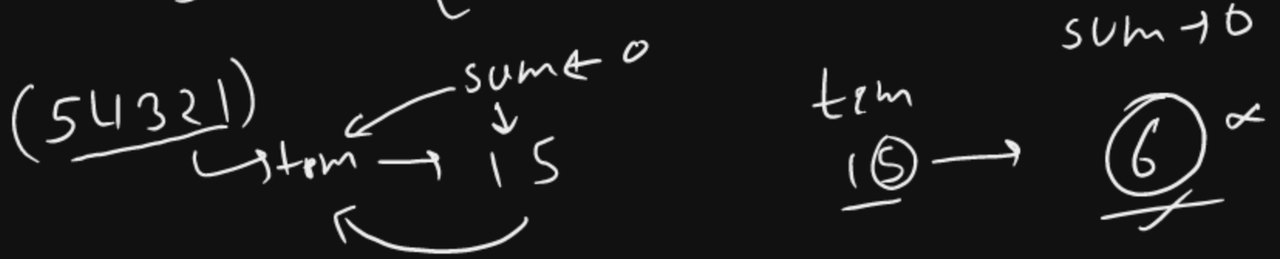
9 → Number
↳ add digits of the number untill the result will become one digit

⇒ Pseudo code

```
1. int n; sum=0
2. while([n!=0])
3.     [sum += n%10
4.      n /= 10]
5.
```




```
sum=0;
while(sum>=10)
tem = (sum==0) ? n : [sum]   (sum=0)
    [while(tem != 0)
        sum += tem%10
        tem /= 10]
```

(sum = 6)

Complexity (coding)

↳ (code) → (time)

↳ (Java) High level, statically

↳ (Python) High level, dynamic

↳ (Asm) Low Level

int n = 10

computer

↳ rate of change

[code] → sum for n

LL

1. Asm → 0.001 s

HL, ST

2. Java → 0.01 s

HL, DT

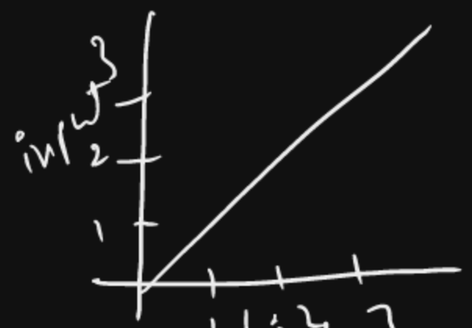
3. Python → 1.025

2.02

↳ 3.03

(rate) time complexity

↳ rate of change of time acc input



(n^2)

→ constraints (n)

[$1 \le n \le 10^{18}$]

→ time limit ⟶ time

[2 sec , 5 sec, 1 sec]

→ [1 sec → 10^8 operation

Operations (worst)

↳ value assign $O(1)$ –

↳ array get $O(1)$ –

↳ update at index $O(1)$ –

✗

worst → for (i → 0 — n) →

→ for (j → 0 — n) →

→ [print (*)] → $O(25), O(3^n), O(4^n), O(n^2)$

→ print ln( ) → $O(n)$

[$O(n \times n) + O(n) + O(n)$]

time rate

$$O(1) < O(\sqrt{n}) < O(n) < O(n \log n) < O(n^2) < O(n!) < O(2^n)$$

$\rightarrow O(n^3)$, $O(\sqrt{n} \log n)$ $\left(\lceil \frac{n}{3} \rceil\right)$

$O(\log n)$

```
→  for (i = 0 → n)  → O(n)
→     for (j = (1) → n)  → O(n) + O(n-1) + O(n-2) .... O(n-n)
→        print()  →  O(n x logn)
→  pln()
```