L2- Time and Space Complexity

What's time complexity?

$\boxed{T.C \;!= \text{time taken}}$

Code → old windo ⇒ 2s

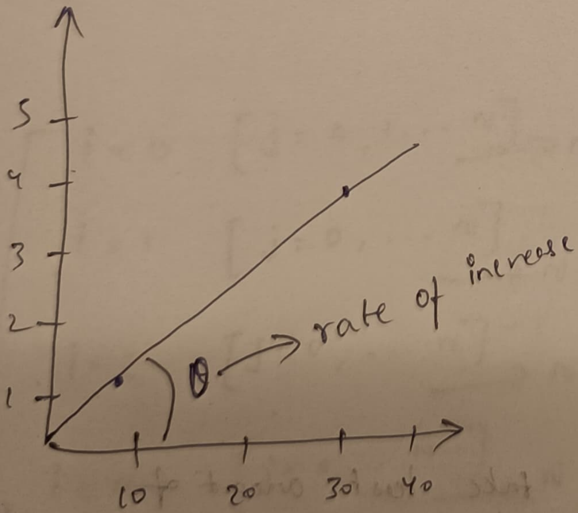
Code → new windo ⇒ 1s

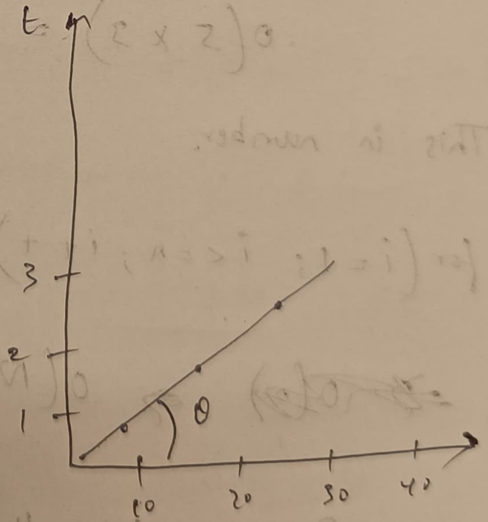
So,

→ The rate at which time taken increases w.r.t input.

old windows

new windows





→ Depending on the Input

→ Big-Oh Notation: O( )
↳ time taken

```
for (int i = 1; i <= 5; i++)       1
{                                   2
                 ⇒ 5 times          3
    cout << "Rizon";                4
                                    5
}
```

total no of steps ⇒ prime big Oh - notiation

Rules for computing T.C

→ Worst Case Scenerio

→ Avoid constants

→ Avoid lower value

So for the above e.g.

3 things are happening for the 5-time

$O(5 \times 3) = O(15)$

This in number.

```
for (i = 1; i <= n; i++)
```

~~O(n)~~ ⇒ $O(N \times 3)$

Best Case
Average Case } 3-Cases
Worst Case

Best Case → When the programe take least amount of time.

Always compute with worst case

Big-Oh (O)

↓

Worst Case

[Upper-Bond]

Theata (Θ)

↓

[Average Complexity]

Omega (Ω)

↓

Best Case

[Lower-Bond]

Q

```
for (i = 0; i < N; i ++)   ⟶ N- times
  {
     for (j = 0; j < N; j ++)   ⟶ N- times
     {
            // Block of Code

     }

  }
```

$i = 0 \quad [j = 0, 1 \ldots N] \rightarrow N$

$i = 1 \quad [j = 0, \ldots N] \rightarrow N$

$i = 2 \quad [j = 0, \ldots N] \rightarrow N$

$i = N-1 \quad [j = 0 \ldots N-1] \rightarrow N$

N

N

$N \times N \Rightarrow O(N^2)$

Q2

```
for (i=0; i<N; i++) ⟶ N times
{ for (j=0; j<=i; j++)
    {
        // Block of code
    }
}
```

$i=0 \quad \{j=0\}$ — 1

$i=$ ~~2~~$1 \quad \{j=0,1\}$ — 2

$i=2 \quad \{j=0,1,2\}$ — 3

⋮

$i=n-1 \quad \{j=0,1,\dots n-1\}$ — 4

$(1+2+3+4+\dots n-1)$

↳ Sum of n natural number

$$\Rightarrow \frac{N(N+1)}{2} = \frac{N^2}{2} + \boxed{\frac{N}{2}} \rightarrow \text{avoid}$$

$$O\left(\frac{N^2}{2}\right) \approx O(N^2)$$

Space Complexity

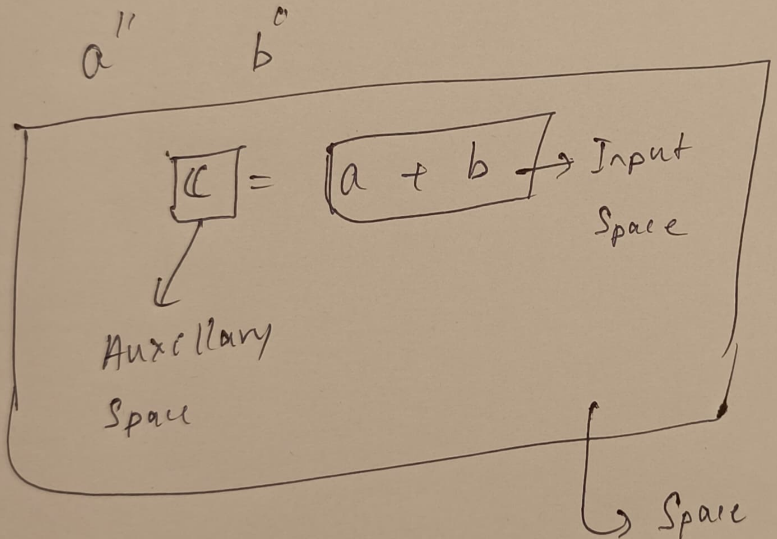
→ Memory Space

→ Big-O Notation

Auxillary Space + Input Space

↳ Space to take for solving the problem

↳ The space you take to store the input.



int a [n]; O(N) //Space

* Never do anything to input, always take extra variable.

for e.g. to add two nos

a b

b = a+b ✗ (Don't do this)

instead do this

c = a+b.