

Time Complexity

Time taken \propto Time complexity

old machine

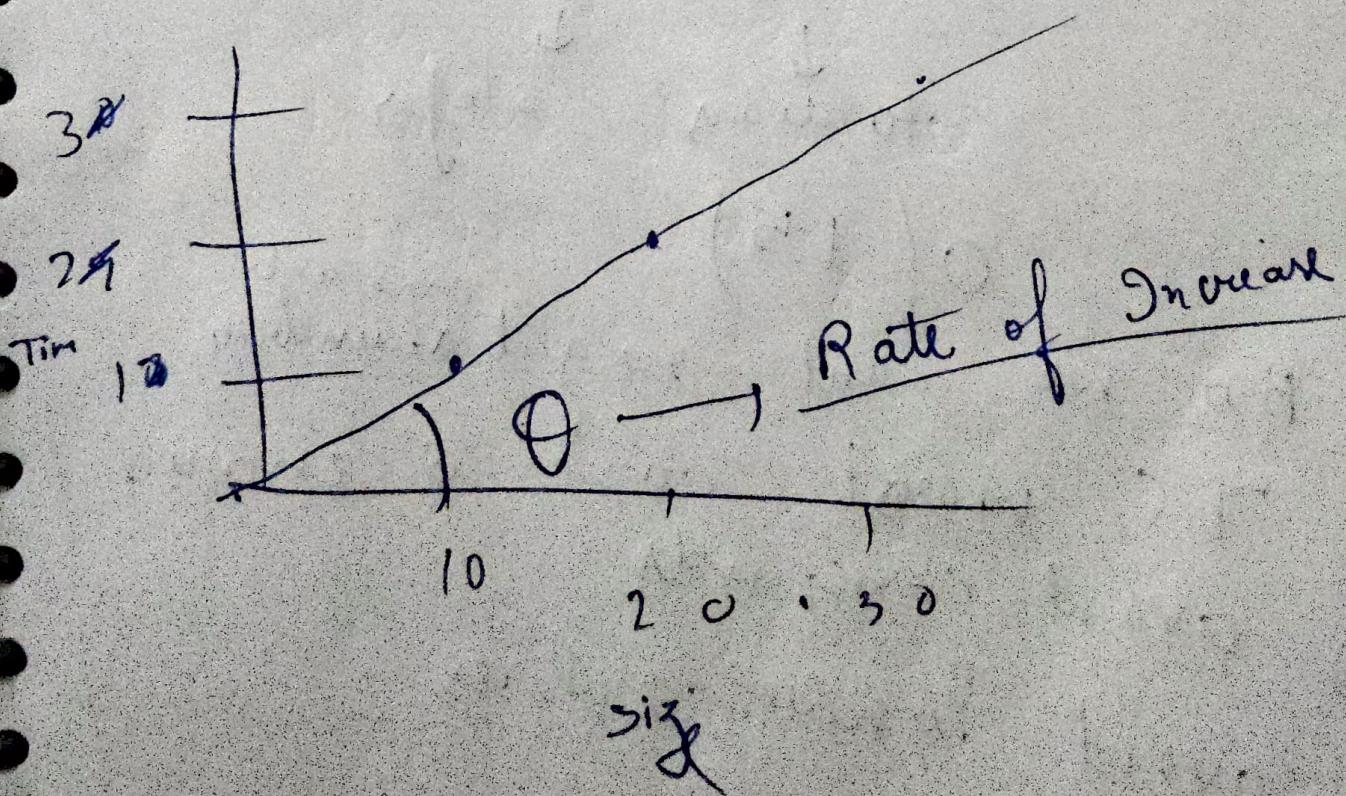
10 s

New machine

3 s

Hence Time taken \propto TC

Time Complexity \rightarrow The rate at which time taken increases wrt input size



Computation of Time Complexity
not sec, not min → but

$O(\rightarrow)$ number of steps

for (int i=0; i < 5; i++)
 cout << i << "(n);

}

5×3
↓ ↓
iterations steps

$O(15)$

But we can't neglect ^{constant} number

- 1) TC, worst case scenario
- 2) avoid constants
- 3) avoid lower values

if the same loop was for,
for (int i=0 ; i < n ; i++)
{
 cout << i << endl;
}

$O(3 \cdot n)$

1) Always consider the worst time complexity

eg
if (①) return ①
if (①) return ①
if (①) ①
else { ① } ①

But $O(2) = \text{TC of if block}$
if execute ~~if~~ statement
worst ③ if conditions & 1 else
block $\rightarrow O(4)$
so T.C $\rightarrow O(4)$ ✓

$$O(4N^3 + 3N^2 + 8) \quad N = 10^5$$

$$O(4(10^{15}) \cdot 3(10)^{10} + 8)$$



$$O(4 \cdot 10^{15})$$

no significant
Avoid constants

no significant
Avoid lower values
(at high
 N)

Ans - $\text{for } (\text{int } i=0; i < N; i++)$
 $\quad \quad \quad \text{for } (\text{int } j=0; j < N; j++)$

Code block → {
 }

{ }

{ }

$i=0 \quad j \in \{1, 2, 3, 4, 5, 6, \dots, N-1\}$

$i=1 \quad j \in \{1, 2, 3, 4, 5, \dots, N-1\}$

$i=2 \quad j \in \{1, 2, 3, 4, 5, \dots, N-1\}$

$i = N-1 = \{ 1, 2, 3, \dots, N-1 \}$

$$N^2 - \overline{1}^2 \approx \underline{\underline{N^2}}$$

Ques- 2

```
for (int i=0; i < N; i++) {  
    for (int j=0; j <= i; j++) {  
        code block  
    }  
}
```

$i=0 \quad j=\{0\}^1$

$i=1 \quad j=\{0, 1\}^2$

$i=2 \quad j=\{0, 1, 2\}^3$

$i=3 \quad j=\{0, 1, 2, 3\}^4$

$i=N-1 \quad j=\{0, 1, 2, 3, \dots, N-1\}^N$

$1 + 2 + 3 + 4 \dots N-1$
If i lab (N-1) till

$$\frac{(N-1)(N+1)}{2}$$

2

$$\left[\frac{N^2}{2} - \frac{N}{2} \right] = \approx 0 \quad \approx O(N^2)$$
$$\approx \frac{N^2}{2}$$

$$\frac{N(N+1)}{2}$$
$$\frac{N^2}{2} + \frac{N}{2}$$
$$\approx N^2$$

Space complexity \rightarrow
Memory space your program
takes

Necessary Space + Input Space

```
int a, b ; → Input space  
cin >> a ;  
cin >> b ;  
int c = a + b ; → Auxillary space  
 $O(3)$ 
```

Do not temper with the input data
You can take some extra variable /
array which instead of $O(N)$ change
your space complexity to $O(2N)$

but dont do

Input a = 1 ;
b = 2 ;

Output →
 $b = b + a ; \times \times$

don't change input state -