Time & Space Complexity

P → Solve

Solve → S_1, S_2, S_3, S_4

S_1: 2s

S_2: 5s

S_3: 10s

S_4: 11s

Q. Sum of n number

```
for (i=1; i<=n; i++){
    sum += i;
}
```

New

64 Gb Ram

2 sec

10 sec

n, n^2, n^3

$n = 10 \uparrow$

$n = 100000 \uparrow$

Old

4 Gb Ram

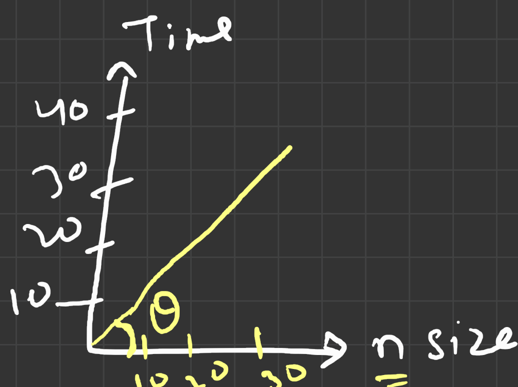
$n = 10$

10 sec

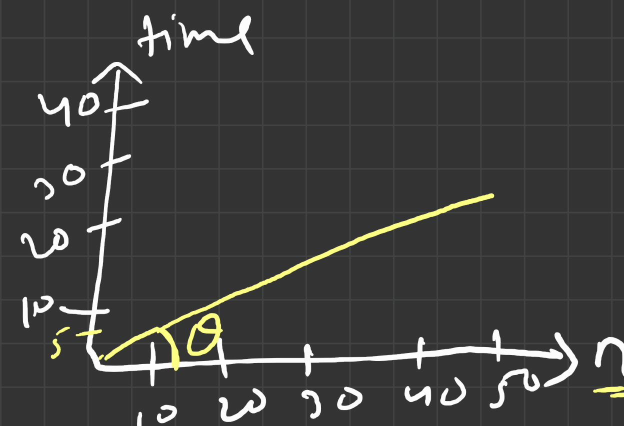
100 sec

Time Complexity ≠ Taken taken

Time Complexity:- It is the total time taken by an algoritm to run as a function of length of the input



old



Best case, Worst Case & Avg case

3	4	30
10		100
7		70
20	avg case	200
40	200	400

best case

worst case

50000

5000

3 termonology ?

Big (O)	Ω (Omega)	θ
worst case	Best case	Avg case
		θ
$O(n)$	$\Omega(n)$	$\theta(n)$
$O(n^2)$		
$O(n^3)$		

$(i = 1 \; , \; i <= n \; ; \; i++) \{$

1 3 cout << "*"

}

$1 + 2 + 3 \;\; --- \;\; n$

$4 + 3 + 3 \;\; --- \;\; 3$

$\underbrace{1 + 3 + 3 + 3 \;\; --- + 3}$

$\boxed{3n + 1}$

constant term → ignore

in n term we take highest value

$O(n)$

$$\cancel{2}N^3 + \cancel{3}N^2 + \cancel{48}N + \cancel{64}$$

$$N^3 + N^2 + N$$

✓ ✗ ✗

$$N^3$$

$$N^3 + \log N$$

$$N^3$$

Made with Goodnotes

```
for (i=0; i<n; i++){
    if (arr[i] == x){
        cout << " this is result"
        break;
    }
```

$$avg = \frac{1+2+3 \ldots +n}{n}$$

$$\frac{\frac{n}{2} \times (n+1)}{\cancel{n}}$$

$n+1$

n

$O(10)$

$O(1)$

$\boxed{x=5}$

best case

5	6	8	7	12

cout << $\frac{n(n+1)}{2}$

$O(1)$

```
for (i = 1;  i <= n;  i++) {            → n  }
    for (j = 1;  j <= n;  j++) {        → n  }  n^2

        cout << '*'
    }
}
```

i = 1
j = 1 to n

i = 2
j = 1 to n

i = 3
j = 1 to n

* print n times

$n + n + n + n \;----\; + n$

$(n)n = n^2$

$\boxed{O(n^2)}$

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

    cout << '*'
```

$i=1$
$j=1$ to 1
$*$ print 1 time

$i=2$
$j=1$ to 2
2 $*$ print

$i=3$
$j=1$ to 3
3 $*$

--- $i=n$
$j=1$ to n
n $*$

$$\underline{1+2+3+\cdots+n}$$

$$\frac{n\times(n+1)}{2} = \frac{n^2+n}{\cancel{2}}$$

$$n^2+n = O(n^2)$$

Q1 for (i=1 ; i<=n ; i=i×2){

```
cout << '*'
}
```

Q2 for (i=1; i<=n ; i++) {

```
for (j=1 ; j<=i^2 ; j++){
  for (k=1 ; k<= n/2 ; k++){
    cout << '*'
  }
```

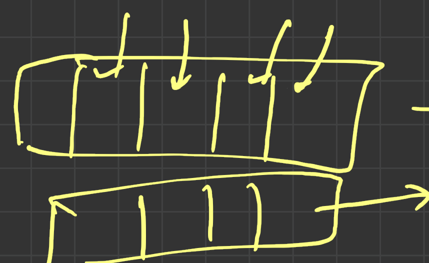
```
for (i=1; i<=n; i++){        → n ✓
  for(j=1; j<=m; j++){       → m ✓
      cout<<'*'
```

$O(\underline{n}+\underline{m})$

Space Complexity:- It is the amount of space taken by an algorithm as a function(fn) of length of i/p.

→ Auxiliary Space

→ Total Space Complexity



→ n

→ data

Auxi $\longrightarrow$ n $\rightarrow O(n)$

Total $\longrightarrow$ n + n $= 2n \rightarrow O(n)$

```
for (i=1; i<n; i++)
    cout << '*'
```

Aux $\rightarrow$ i $= O(1)$

Total $= 1 + i = O(1)$

$O(N!)$ → worst

$O(2^N)$

$O(N^n)$

$O(N^3)$

$O(N^2)$

$O(N)$

$O(N \log n)$

- $O(N)$
- $O(\sqrt{n})$
- $O(\log n)$
- $O(1)$ → Best