Q

```
for (i=0 ; i<N; i++) {
    for(j=0 ; j<N; j++) {
        for (k=0; k<=j ; k++) {
            ...
        }
    }
}
```

$O(N^3)$

→ How to count no of iteration ?

→ How to get Big-O from no. of iteration ?

⇒ Why we need TC ? ⟹

⇒ What is Big-O ? ✓

⇒ How Big-O is related to no. of iteration ? ✓

⇒ What are the advantages of constraints ?

⇒ What is SC ?

arr : [3, 8, 9, -1, 10, 4]

→ [-1, 3, 4, 8, 9, 10]

Anmol

"Sidha Sort"

10sec

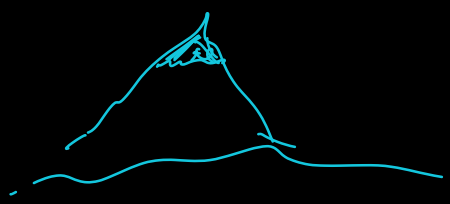
macbook pro

10 sec

Python

C++

6.5 sec



Devyanshu

"Bestone Sort"

1.5 sec

Samsung

Hardware

7sec

C++

Language

C++

7 sec

Environ

Iteration ⇒ Only dependent on algo.

Abhishek
"Start Sort"

$100 \log_{10} N$

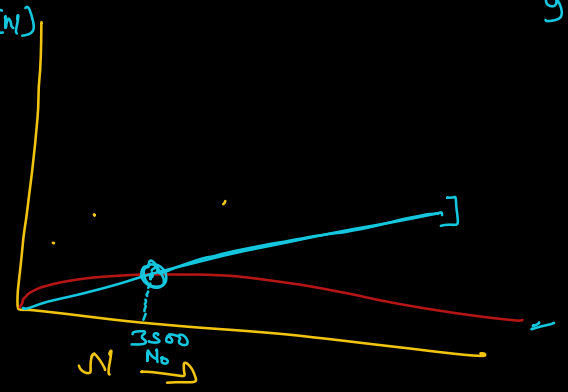
$f(N)$

$\hookrightarrow O(g(N))$

Mukul
"Strong Sort"

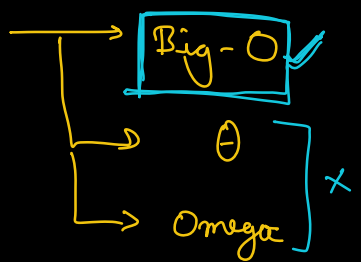
$N/10$

$g(N)$



Asymtotic Notations

$N \to \infty$



Big-O : [Given a function $f(N)$, for $N \geq 0$ it is $O(g(N))$, if there exists two constants c & N_0 such that $f(N) <= c\,g(N) \quad \forall N >= N_0$]

1st year

Sir

$f(N) = N^2 + N$ $\quad N>=0$] No of iteration

$N^2+N \quad <= \quad N^2 + N^2$?

$N^2+N \quad <= \quad 2(N^2)$ $\quad N >= 0$?

$f(N)$ — c — $g(N)$ — N_0

$f(N) \quad <= \quad c\,g(N) \quad N >= N_0$?

$O(N^2)$

Q $\quad f(N) = N^2 + 5$

$N^2+5 \quad <= \quad N^2 + 5N^2$

$N^2+5 \quad <= \quad 6(N^2)$ $\quad N >= 1$

$f(N)$ — c — $g(N)$ — N_0

$f(N) \quad <= \quad c\,g(N) \quad N >= N_0$

$f(N) \rightarrow O(g(N))$

$O(N^2)$

Q $f(N) = N^3 + 8N^2 + N \log N$

$N^3 + 8N^2 + N \log N <= N^3 + 8N^3 + N^3$

$N^3 + 8N^2 + N \log N <= 10N^3$ || $f(N) <= 10N^4$, $f(N) <= 10N^3$, $f(N) <= 10N^{10}$

$f(N) <= cg(N)$

$f(N) \rightarrow O(N^3)$

Q $f(N) = 10N^2 + 10^4 \quad N \geq 0$

$10N^2 + 10^4 <= 10N^2 + 10^4 N^2 \quad N > 0$

$10N^2 + 10^4 <= 10010 N^2 \quad N > 0$

$10N^2 + 10^4$ → $f(N)$; 10010 → c; N^2 → $g(N)$; 0 → N_0

$f(N) \rightarrow O(N^2)$

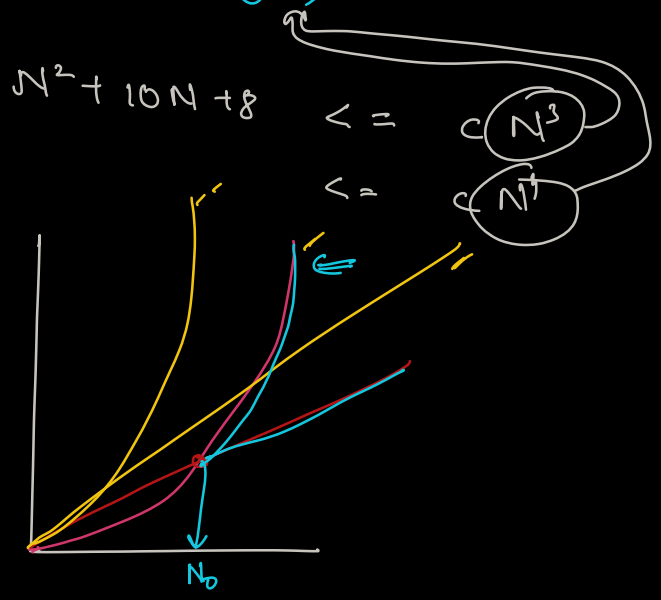
1. Pick only the largest power
2. Remove constants

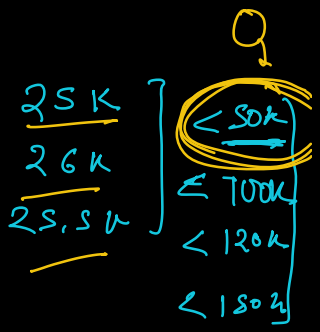
Quiz

$f(N) = N^2 + 10N + 8$ → $O(N^2)$, $O(N^3)$, $O(N^4)$

$f(N) <= c * g(N)$ for $N >= N_0$

$f(N)$ → $O(g(N))$





Distances from your home

Hospital :	2 Km	⇒ Walk / Bycycle
Cinema :	4 Km	⇒ Walk / Bycycle
Airport :	25 Km	⇒ cab
Kashmir :	1000 Km	⇒ Train / flight
USA :	13000 Km	⇒ flight
The Moon :	3,84,400 Km	⇒ Rocket

Earth → Moon : Home → Airport → USA airport → Nasa → Moon

- N^2
- $10\,N^2$
- $8N^2 + 6N + 25$
- $16N^2 + 4gN$

$N \to \infty$

$O(N^2)$

- $N^3 + N^2 + N + 1$

N^3

$6N^3 + 25N^2$

$25N^3 + 100$

$O(N^3)$

Break till 10:34 pm

List → find(Max)

=> $O(N)$

SDE-2 Sorted, rt A[0]

$O(N \log N)$

Q

```
for (i = 1;  i<=N;  i++){
    for (j = 1 ; j<= N; j++){
        for ( k= 1;  K<=N ; k++) {
            ---
        }
    }
}
```

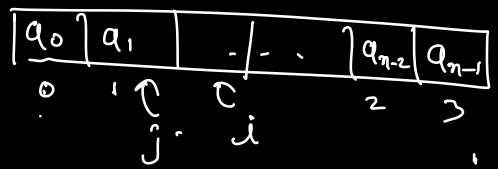
$\Rightarrow O(N^3)$

Q.

```
void fn( a[], N){                    a[i] > 0 ∀ i
    j = 0;
    for( i= 0;  i<N ; i++){
        while ( j < N  &&  a[i] < a[j]) {        → i != j
            j++;
        }
    }
}
```

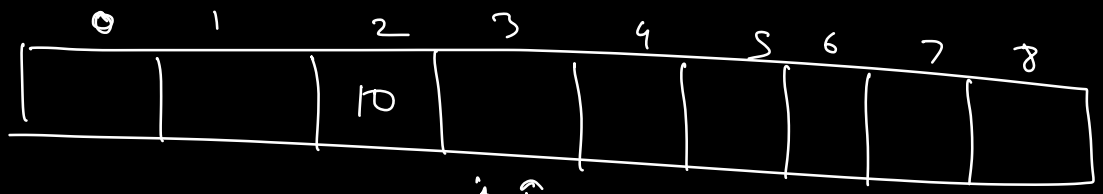
N × 0

i	j	No of iterations
0	0	
1	0 → 1	



$a_0 = a_0$

$a_1 < a_0$



j i

$a[i] = 10$

$a[j] = 10$

$a[i] < a[j]$ ×

$i = 0 \rightarrow 0$

$i = 1 \rightarrow 1$

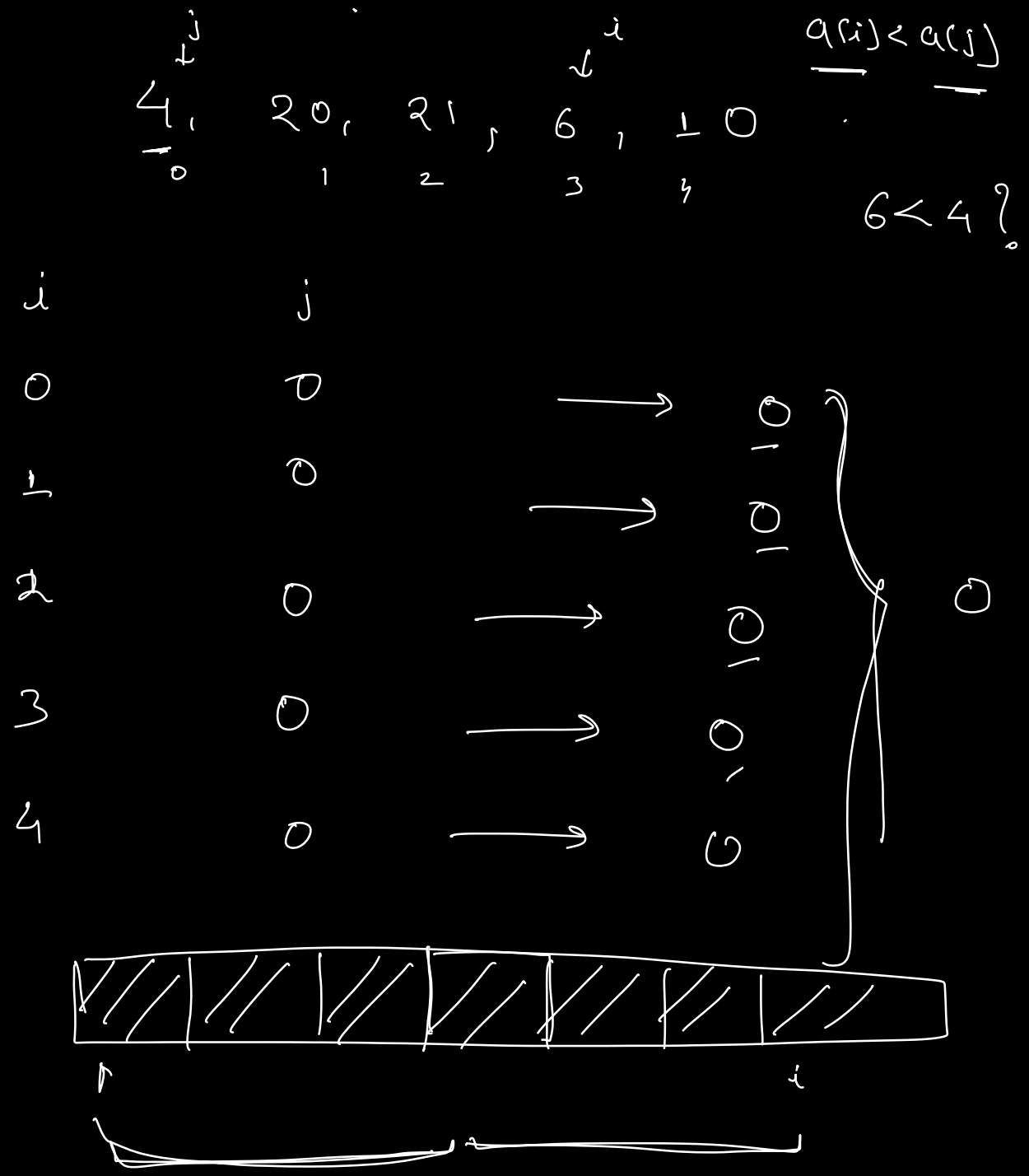
$i = 2 \rightarrow 1$

$i = 3 \rightarrow 1$

$i = 4 \rightarrow 1$

$\vdots$

$O(N)$



O(N)

Space Complexity

1 int ⇒ 4 Byte

1 long ⇒ 8 Bytes

```
void fn(int N) {
    int  a = N*N;        // 4 Bytes
    long b = N*N*N;      // 8 Bytes
    int ar[N] = new int[N];   // 4N Bytes
    int matrix[N][N];
}
```

12 Bytes. O(1)

Space = 12B + 4NB O(N) =

Space = 12 + 4N + N² ⇒ O(N²)

```
fn(int N, M) {
    int ar[N][M]
}
```

O(NM)

CP

Spoj
Codechef
Codeforces
:
etc.

1s / 2s ?

Online Judge

TLE ⇒ Time limit exceeded

1 Ghz ⟶ 10^9 machine level instruction

1 iteration ⟶ 10 MLI (approx) ⟶ 100MHz (approx)

1 GHz Processor can perform ⟹ $(10^7 - 10^8)$ iterations in 1 sec

```
for (i=0; i<N; i++){
    {
    }
}
```

in 1 sec ⟶ 10^7 - 10^8

$O(N)$ ⟹

$N \Rightarrow 10^2$

Solve a Problem

→ Read the problem statement
⇒ Analyse Constraints
→ Logic
→ Write Code
→ Calculate TC

$O(N^3)$

Importance of Constraints

$O(N^3)$ algo

$1 \le N \le 10^9$

$(10^9)^3 = 10^{12}$

10^8

$1 \le N \le 100$

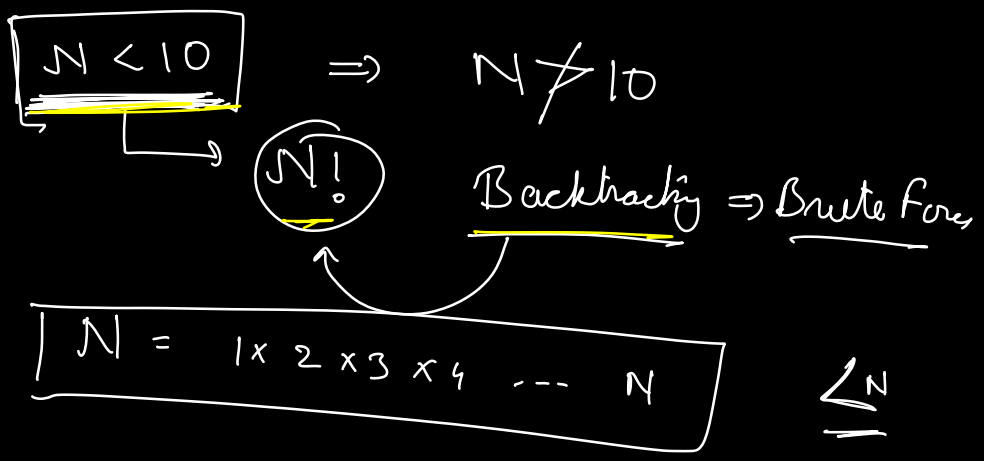
$(100)^3 = 10^6$

TC	$\log N$	$\sqrt{N}$	$N \log N$	N^2	2^N	$N!$
$N = 10^6$	$\log_2 10^6$	$\sqrt{10^6}$	$10^6 \log_2 10^6$	10^{12}	$N = 20$	$N = 10$
	$\downarrow$				2^{20}	3×10^7
	20	10^3	2×10^7	x	10^6	
			$2 \times 10 \times 10^6$			
			2×10^7			

$2^{10} = 1024 \approx 1000$

$2^{30} \approx 10^9$

$2^{20} \approx 10^6$



Q Given an array.
Find sum of all the elements.?

```
int fn(arr(), N) {
    int S = 0;
    for (i=0; i<N; i++){
        S = S + arr[i];
    }
    ret S;
}
```

$1 \le N \le 10^5$

$1 \le arr[i] \le 10^6$

| 10^6 | 10^6 | --- | 10^6 |

$S = 10^6 \times 10^5$

$S = 10^{11}$ ⇐ int ?

Overflow

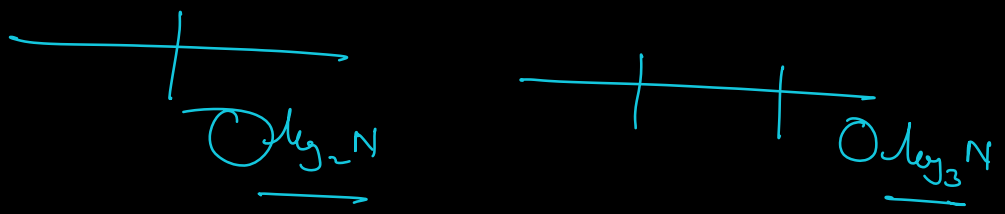
TC: O(N) ⟹ 10^5 ✓

SC: O(N)

Extra/Auxiliary Space: O(1)] ⟶ ✓

$\log_b N = c \log_d N$

$\log_2 N \approx \log_3 N \approx \log_{10} N$



- AP
- GP
- log
- HCF - LCM
- Prime No

9 - 12 th

30 min (Max)

$N \leq 10^6$

TC : $O(N!)$ $(10^6!) \rightarrow$ iterate

→ Optimize

$N = 100$ ✗

$N \leq 10$ ✓

CP ⇒ Maths ⇒ [Permutation, Probability, Geometry, Convex Hull] ✗

↓

Sports