$arr[] = \{3, 1, -1, 6, 4\}$

$\downarrow$

$\{-1, 1, 3, 4, 6\}$

Venkat

[Omicron]

10 s

MAC book Pro

10s

Python

$\downarrow$

C++

6.5 s

7.1 sec

Aparna

[AjClassic]

15 s

Windows

$\Downarrow$

MAC book Pro

7s

C++

7s

6.9 sec

For some other test case

$\downarrow$

5.3 s

$\downarrow$

11 s

J.S $\rightarrow$ C++

NO%

Execution time :

HW
Language
place
selection i/p
⋮

Not a correct measure to compare algorithms.

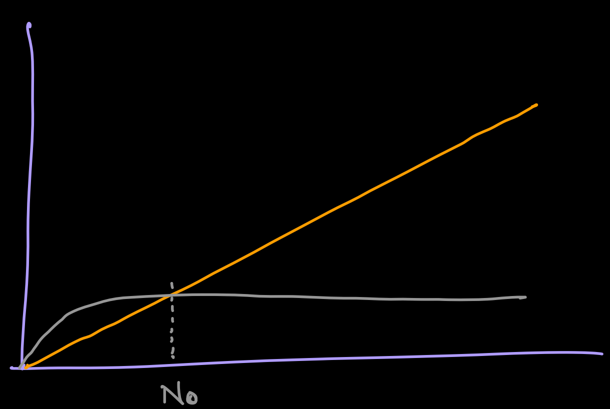
```
for( i=1; i<=N; i++) {
    |
}
```

i: N iterations.

#Iterations:

$O(\log N) < O(N)$

$\boxed{100 \log N}$ $\frac{N}{2}$



Asymptotic Analysis - BIG O

observing behavior / growth of Algorithms for large inputs.

- Big O (O)
- Omega (Ω)
- Theta (θ)

} Reading material.

Big O

1. # iterations
2. Neglect lower order terms
3. Neglect the constants from higher order term.

Contribution of lower order term to the # iterations

	N^2	$+10N$	Total
$N = 10^2$	10^4	10^3	$10^4 + 10^3$
$N = 10^4$	10^8	10^5	$10^8 + 10^5$

$$\frac{10^3}{10^4 + 10^3} \times 100 = 9\%$$

$$\frac{10^5}{10^8 + 10^5} \times 100 = 0.1\%$$

$N = 10^5$ $\quad 10^{10} \quad 10^6 \quad 10^{10} + 10^6$

$$\frac{10^6}{10^{10} + 10^6} \times 100 = 0.01\%$$

Home $\xrightarrow{384000 \text{ Km}}$ MOON

Home $\xrightarrow[?]{X}$ Airport $\xrightarrow[?]{X}$ USA Airport $\xrightarrow[?]{X}$ SpaceX $\xrightarrow[?]{X}$ Moon

Yogesh

N^2

Kunal

$10 \times N$

For large value of N, Kunal's algo is better

$N * N$ $\qquad\qquad 10 * N$

when $N > 10$
Kunal's algo always perform better

$N * N$ $10^4 * N$

$N > 10^4$

Kunal's algo will perform better.

N^2 vs N

Issues:

Algo 1	Algo 2
$10^3 N$	N^2
$O(N)$	$O(N^2)$

Can we simply claim that Algo1 is better?

$O(N)$ $O(N^2)$

N	Algo1		Algo2
	$10^3 N$		N^2
10	10^4	$>$	10^2

Algo2 is better

10^2	10^5	$>$	10^4	Algo2 is better
10^3	10^6	$=$	10^6	Both are same
10^3+1	$10^3(10^3+1)$	$<$	$(10^3+1)(10^3+1)$	Algo1 is better
10^3+2	$10^3(10^3+2)$	$<$	$((10^3+2)(10^3+2))$	"

Doesn't say anything w.r.t the smaller inputs.

Algo 1

$N^2 + \cancel{10N}$

$1 * N^2$

$O(N^2)$

Algo 2

$100N^2$

$\boxed{100}N^2$ ✓

N^2

$O(N^2)$

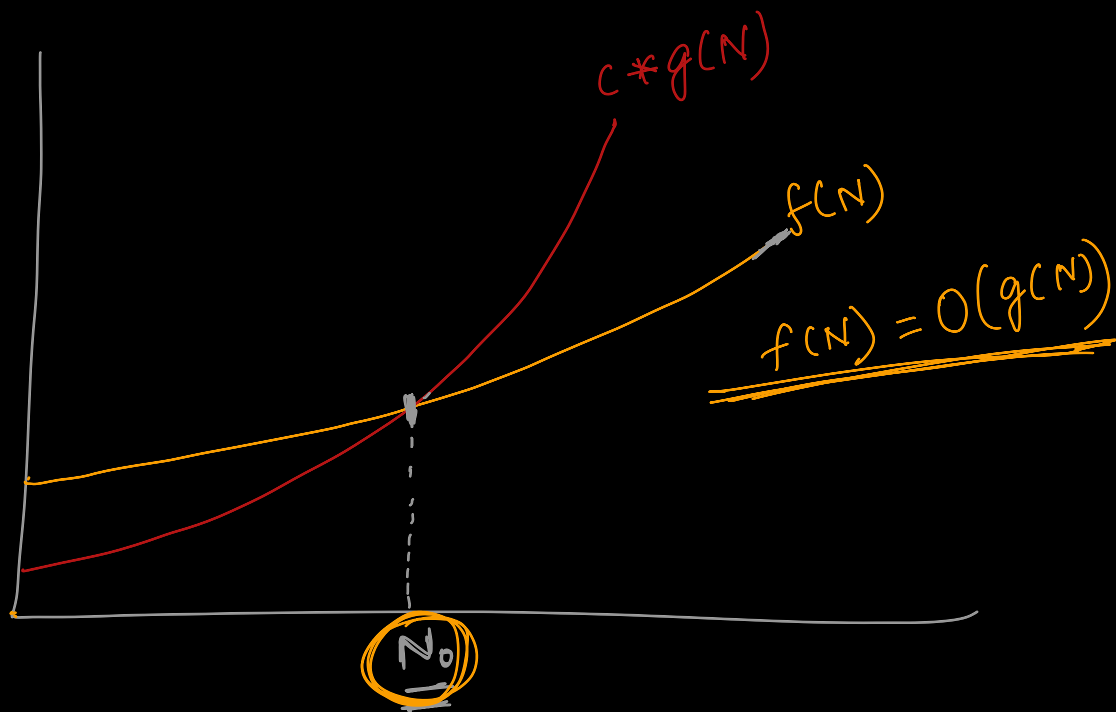
Mathematical Def. of Big O

All iterations are function of input. [say N].

$f(N) = O(g(N))$ if there exists two constants (c & N_0) s.t $c * g(N) >= f(N)$ $\forall N >= N_0$.

→ $f(N)$ is a function

→ $g(N)$ is a function



After N_0,

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

Red line is acting as upper bound for f(N)

$f(N) = O(g(N))$

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

$\boxed{f(N) = N^2 + 5N}$ $\quad$ $g(N) = 2N^2$

$f(N) = O(g(N))$

N	$N^2 + 5N$		$2N^2$
3	24	>	18
4	36	>	32
5	50	=	50
6	66	<	72
7	84	<	98
⋮			⋮

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

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

$g(N) = N^2$

$f(N) = O(N^2)$

N	$f(N)$: $N^2 + 5N$		$g(N)$: $10N^2$
$N=0$	0	=	0
$N=1$	6	<	10
$N=2$	14	<	40
$N=3$	24	<	90

$\underline{\underline{N >= 0}}$

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

$$N^2 + 5N <= 10 * N^2 \quad \forall\, N >= 0$$

$$g(N) = N^2$$

$$f(N) = O(g(N))$$
$$= O(N^2)$$

N	f(N)		g(N)
	N^2+10N		$100N$
100	$10^4 + 10^3$	$>$	10^4
101	$(101)^2 + 10\times(101)$	$>$	$(100)(101)$
102	$(102)^2 + 10\times(102)$	$>$	$(100)(102)$
⋮	⋮		⋮

since $f(N) > c * g(N) \;\forall\; N$

$\therefore$ we can't have

$f(N) = O(g(N))$

i.e $N^2+10N \neq O(N)$

$f(N) = 10N^3 + 10^6$ $\qquad$ $g(N) = 11N^3$

$f(N) = 10N^2$ $\qquad$ $g(N) = N^3$

N	$10N^2$		N^3
9	810	>	729
10	1000	=	1000
11	1210	<	1331
12		<	
⋮		<	

$$f(N) = O(g(N))$$
$$= O(N^3)$$

Running speed is < light < train < bolt < 6 Km/h

tigther

Space Complexity

int → 4 Bytes
long → 8 Bytes
double → 8 Bytes

```
void func(N){
    int a,b,c
    long d
    double e
    print(a*b*c*d)
}
```

$4 \times 3 + 8 + 8$

$= \underline{28}$ Bytes

O(1) since the space doesn't not grow w.r.t the input size (N).

```
void func(N){
    int a,b,c
    long d
    double c
    int arr[N]
}
```

$3 \times 4 + 8 + 8 + N \times 4$

$28 + 4N$

$\underline{O(N)}$ → space Comp.

```
void func (N) {
    int a, b, c
    long d
    double c
    int arr[N]
    int mat[N][N]
}
```

$3 \times 4 + 8 + 8 + 4 \times N + N \times N \times 4$

$28 + 4N + 4N^2$

$O(N^2)$

Given N array elements, calc sum of all array ele.

N   {3, 7, 2, -1, 0}

```
int sum( int arr[], int N){
    int s = 0
    for( i=0; i<N; i++) {
        s = s + arr[i]
    }
}
```

4

N x 4

4

4

$\boxed{4 \times N} + 12$

already give

extra auxiliary

O(1)

SC : extra space
space taken

```
int pref(int arr[], int N){    x
    int sum[N]                 -> N x 4
    sum[0] = ar[0]
    for(i=1; i<N; i++){        -> 4
        sum[i] = sum[i-1] + arr[i]
    }
}
```

$4 + 4 \times N$

$= O(N)$.

10 B

VIKAS

AfNAN

10 B + 4 Bytes

10B + 2 Bytes.

```
void fun(int arr[], int N; int K){
    for(i=0; i<N; i++){
        if(arr[i] == K){
            return true
        }
    }
    return false
}
```

Best T.C

$O(1)$

Worst Case

$O(N)$

Average Case

Dhananjaya

Panic Algo

Best $O(1)$

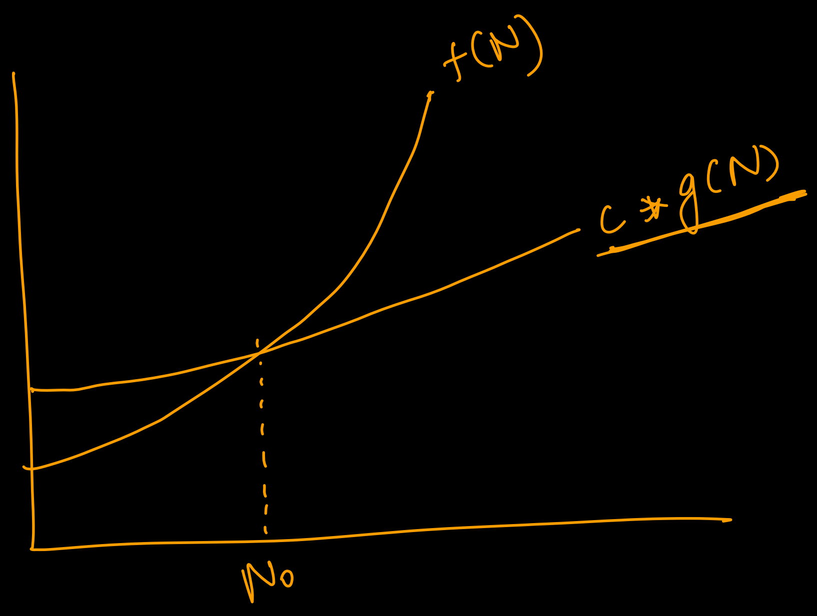
:

Worst $\boxed{O(N^2)}$ BIG O

Omega $\Longrightarrow$ (Best case) $\Longrightarrow$ gives lower bound

$f(N) = \Omega(g(N))$ if there exists two constants c & N_0 s.t

$$f(N) >= c * g(N) \quad \forall N >= N_0$$

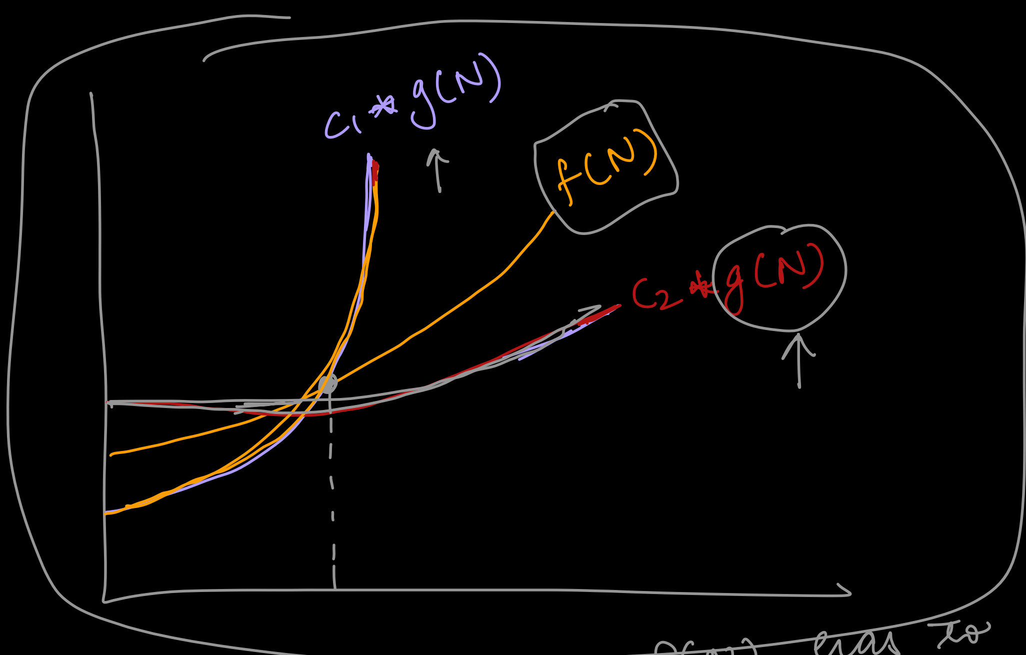


Theta

$f(N) = \Theta(g(N))$ if there exists three constants c_1, c_2, N_0 s.t

$$\underbrace{c_1 * g(N)}_{\text{lower bound}} <= f(N) <= \underbrace{c_2 * g(N)}_{\text{upper bound}} \quad \forall N >= N_0$$

g(N) is the tight bound to f(N)



f(N) has to be equal to g(N) ?

Merge Sort =

Best worst Avg → $\boxed{\Theta(n \log_2 N)}$

```
for (i = n; i > 0; i /= 2){
    if(i % 2 == 0){
        for (j = 1; j <= n^2; j += 2){
            // O(1)
        }
    }
}
```

$1 - n^2$

$\frac{n^2+1}{2}$

$1 - n$

$\frac{n+1}{2}$

i	j	iter
N	$[1 - n^2]$	$\frac{n^2+1}{2}$
$N/2$	$[1 - n^2]$	$\frac{n^2+1}{2}$
$N/4$	$[1 - n^2]$	$\frac{n^2+1}{2}$
⋮		⋮
1		$\frac{n^2+1}{2}$

$\log_2 n \times \left[\frac{n^2+1}{4}\right]$

$\boxed{O(n^2 \log_2 n)}$