- TC / SC ?
- Asymptotic Analysis
- Big O notation
- TLE

---

1) Given 10^5 ints. Sort them in ASC order!

Soumen

python 15 sec [Mac]

[C++] 8 sec [Antartica]

Aditya

20 sec [old laptop]

10 sec [MAC] C++ [Volcano]

8 sec [Antarctica]

Execution time is not a good factor to judge algo's.

└→ SW + HW + External factors

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

}

→ # ops/ # it → N

which does not change with any external factor!

SORT NUMBERS →

	Heman	Aditya
# ops →	$100 \log_2 N$	$N/10$
N=32	$100 \times \log_2 32$	$32/10$
	100×5	~ 3.2 ops
	= 500 ops	
N=64	$100 \times \log_2 64$	$64/10$
	100×6	~ 6.4 ops
	= 600 ops	

$N = 2^{20}$ $\quad 100 \log_2 2^{20}$ $\quad 2^{20}/10$

$2^{20} \simeq 10^6$ $\quad 100 \times 20$ $\quad 10^6/10$

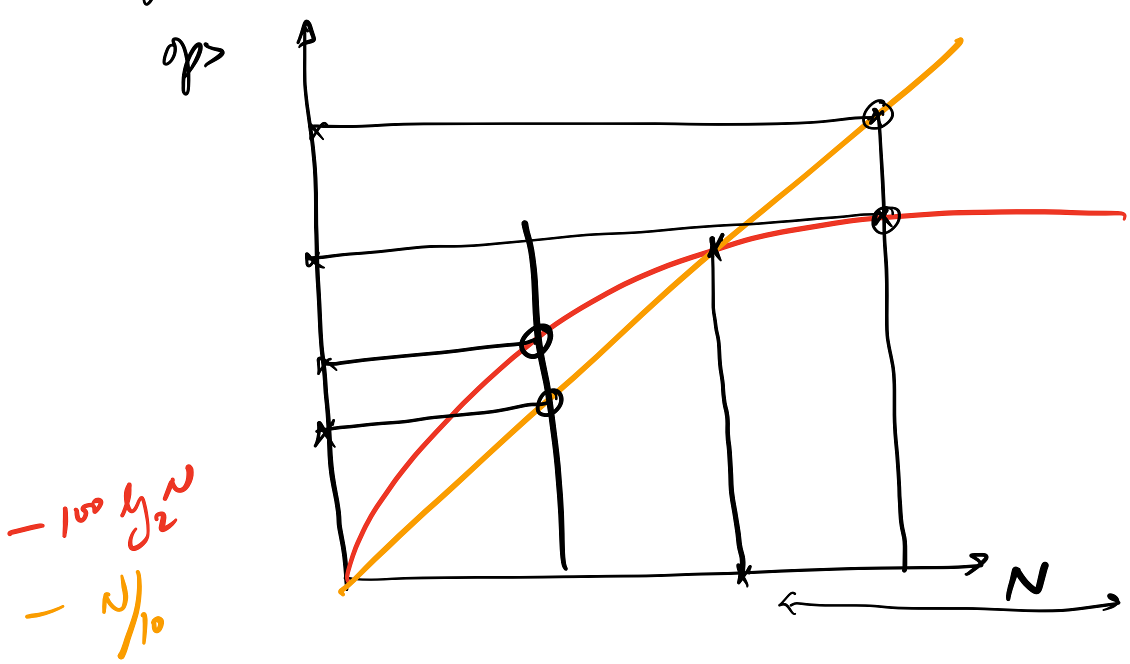
2000 ops $\quad \sim 10^5$ ops

$N = 2^{40}$ $\quad 100 \times 2^{40}$ $\quad 2^{40}/10$

$2^{40} \simeq 10^{12}$ $\quad 4000$ $\quad \sim 10^{12}/10$

$\sim 10^{11}$

($100 \log_2 N$) would perform better for larger values of N, compared to ($N/10$)



⊙ Algo 1, Algo 2, Algo 3 . . . ——— ?

Hotstar	Youtube	Google Search
10^7	10^9	$\sim 10^9$

Input values to algo's are generally larger.

$N = 10^9$

N	N^2
10^9 ✓	$(10^9)^2$
	10^{18}

⊙ Asymptotic Analysis

↳ Used to judge the performance of an algorithm for large input sizes!

$\boxed{10^8 \text{ ops} \rightarrow 1 \text{ sec}}$

$N = 10^8$

N

10^8 ops

↓

1 sec

$N\sqrt{N}$

$10^8\sqrt{10^8}$

$10^8 \times 10^4$

10^{12} ops

10^8 ops $\rightarrow$ 1 sec

1 op $\rightarrow \frac{1}{10^8}$ sec

10^{12} ops $\rightarrow \frac{10^{12}}{10^8}$ sec

10^4 sec

~ 3hrs ✓

N^2

$(10^8)^2$

10^{16} ops

10^{16} ops $\rightarrow \frac{1}{10^8} \times 10^{16}$ sec

$\rightarrow 10^8$ sec

3 0000 hrs

> 3 yrs

① Big O Notation

1) Calculate the no. of ops w.r.t. input
2) Neglect the lower order terms
3) Neglect the constant coefficients!

Input → N

```
cnt = 0, n = 0;
for ( i = 0; i<N; i++) {
    cnt++;
    n--;
}
```

$\sim 2 + 1 + 4N$ ops

$\sim 4N + 3$ ops

$\underbrace{4N^1}_{1} + \underbrace{3N^0}_{0}$ → neglect

neglect ← 4 N

$\boxed{TC = O(N)}$

N

\# ops $\rightarrow$ $f(N) = 10N^2 + 50N + 6$

2   1   0

$10N^2$

N^2

$\boxed{TC = O(N^2)}$

Input $\rightarrow N$

\#ops $\rightarrow$ $f(N) = \underbrace{600N^3}_{3} + \underbrace{5N^2}_{2} + \underbrace{40N\sqrt{N}}_{1.5} + 46\log N$

$600N^3$

N^3

$\boxed{TC = O(N^3)}$

⊙ I/P $\rightarrow N$

$f(N) = 60\underbrace{N^3\log N} + 40N^3$

$\boxed{TC : O(N^3\log N)}$

⊙ $\# \text{ops}$ $\xrightarrow{}$ I/p → N

$f(N) = 400 N^0$

$\downarrow$

N^0

$TC: O(N^0)$

$\boxed{TC: O(1)}$ → TC : Constant!

⊙ I/p → N, M

$\# \text{ops} \rightarrow f(N, M) \longrightarrow \underbrace{40N^3 + 20N^2}_{} + \underbrace{59M\sqrt{M} + 10M}_{} + 6$

$40N^3 + 59M\sqrt{M}$

$\boxed{TC = O(N^3 + M\sqrt{M})}$

⊙ I/p → N, M

$\# \text{ops} \longrightarrow f(N, M) = 4NM + 5N^2M$

$\boxed{TC: O(N^2M)}$

$N >> M$

$\varphi \longrightarrow NM^2 + N^2M$

$\boxed{TC: O(NM(N+M))}$

I/P → N

ops → $f(N) = N^2 + N$

% contribution of N ?

N = 100 $\quad f(100) = 100^2 + 100$

$= 10000 + 100$

$= 10100$ ops

$\frac{100}{10100} \times 100\%$

$\sim 1\%$

$N = 10^5 \quad f(10^5) = (10^5)^2 + 10^5$

$= (10^{10} + 10^5)$

% contribution of N

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

$10^{-5} \times 100\%$

$10^{-3}\%$

0.001%

$\varphi \longrightarrow N^3 M + N^2 \log M$

TC: $O(N^3 M)$

#

$$2^{10} = 1024 \simeq 10^{3}$$

$$2^{20} \longrightarrow \sim 10^{6}$$

$$2^{30} \longrightarrow \sim 10^{9}$$

$$2^{60} \longrightarrow \sim 10^{18}$$

$$10^{15} \longrightarrow 2^{50}$$

$$10^{9} \times 10^{6}$$

$$2^{30} \times 2^{20} \longrightarrow 2^{50}$$

16, 17, 16.5

$$10^{5} \longrightarrow \sim 2^{16.5}$$

$$10^{3} \times 10^{2}$$

$$2^{10} \quad 2^{6.5}$$

$$2^{16.5}$$

$$\log_2(10^{14})$$

$$\log_2(2^{46})$$

$$\sim 46$$

$$10^{14} \longrightarrow 2^{46}$$

$$10^{9} \times 10^{5}$$

$$2^{30} \times 2^{16..}$$

$$2^{46..}$$

Q Given an arr. find if K exists!

```
for(i=0; i<N; i++){
    if(A[i]==K){
        ret true;
    }
}
ret false;
```

Best Case : 1

Worst Case : ~ 3N

3N

$\boxed{TC: O(N)}$

Q

```
f(N){
    ret (N x 2 x 5);
}
```

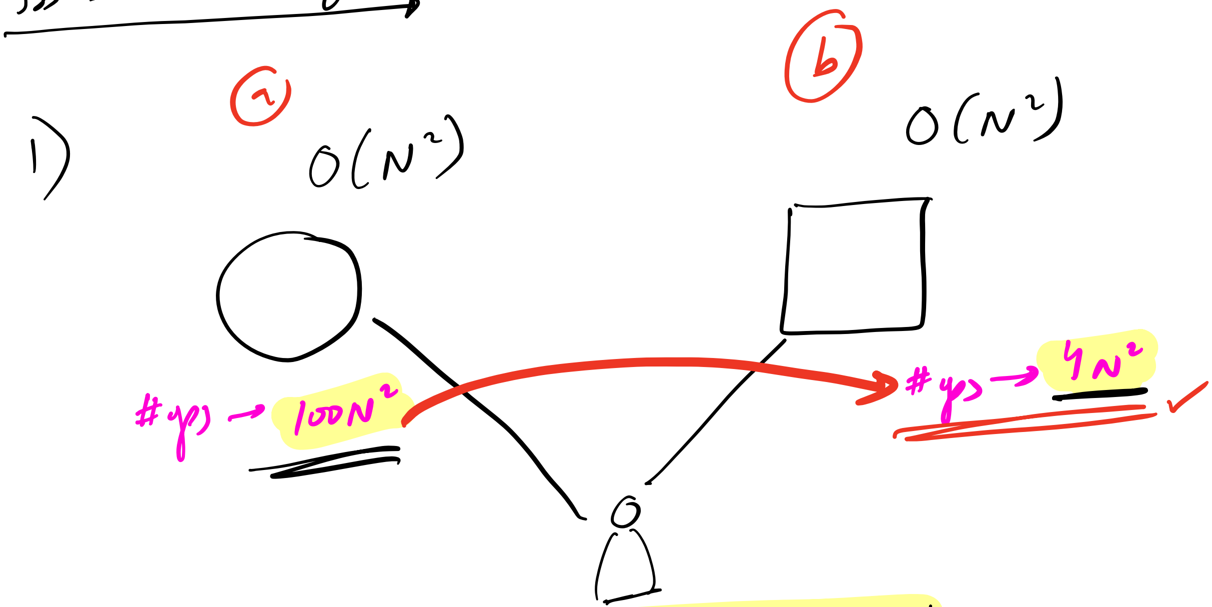
ops → $2 \cdot N^0$

N^0

$\boxed{TC: O(1)}$

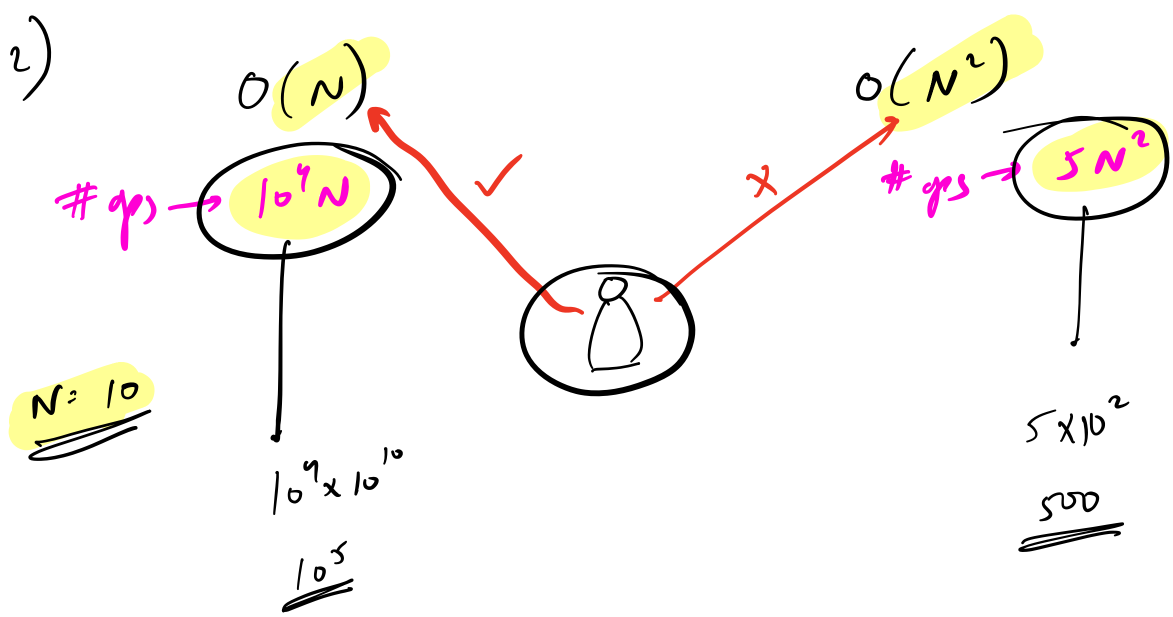
Issues with big O

1)



We cannot get an exact idea of # ops!

2)



→ We cannot decide which algo is better for smaller values of input!

⓪ Space Complexity → Big O

1 int → 4 B
1 double → 8 B
1 long → 8 B

```
f(N) {
    int x = N;          → 4B
    int y = 10;         → 4B
    double z = 10.0;    → 8B
    long v = 40;        → 8B
}
```

$\boxed{24B}$

space $f(N) \to 24\,N^0$

$\downarrow$

N^0

$\boxed{SC = O(1)}$

```
f(N) {
    int x = N;          → 4B
    int y = 10;         → 4B
    double z = 10.0;    → 8B
    int A[N];           → 4N B
}
```

$\to f(N) = 16 + 4N$

$\downarrow$

$\boxed{SC = O(N)}$

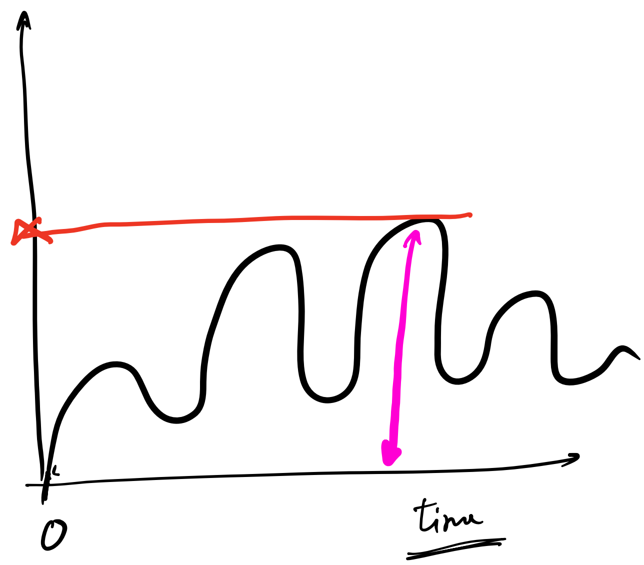
```
f(N){
    int x = N;        → 4B
    int y = 10;       → 4B
    double z = 10.0;  → 8B
    int A[N][N];      → 4N²
}
```

$4N^2 + 16$

$\boxed{SC : O(N^2)}$

Space Complexity

MAX amount of space occupied by your algo at any point



⊙ TLE

Time Limit Exceeded

#ops → $5N^2$

$5 \times (10^6)^2$

5×10^{12} ops

10^8 op → 1 sec

5×10^{12} op → 5×10^4 sec ✗

TLE !

TL : 1 sec / 2 sec

ML : 256 MB

$1 <= N <= 10^6$

$0 <= A[i] <= 10^9$

#ops → $2 N \log_2 N$

$2 \times 10^6 \log_2 10^6$ → 2^{20}

$2 \times 10^6 \times 20$

$= 4 \times 10^7$ ops → < 1 sec ✓

---

f(———) {

N it

4N op → = + - × /

}

1 GHz → 10^9 clock cycles / 1 sec

4 GHz → 4×10^9 —— 1 sec

CPU cycles !

1 sec → 4×10^9 clock cycles

1 op → ~10 clock cycles

4×10^8 ops

$\boxed{1 \text{ sec} \rightarrow 10^8 \text{ ops}}$

```
f(N) {
  ret 5N + 3;
}
```

ops → 2 N^0

$O(1)$