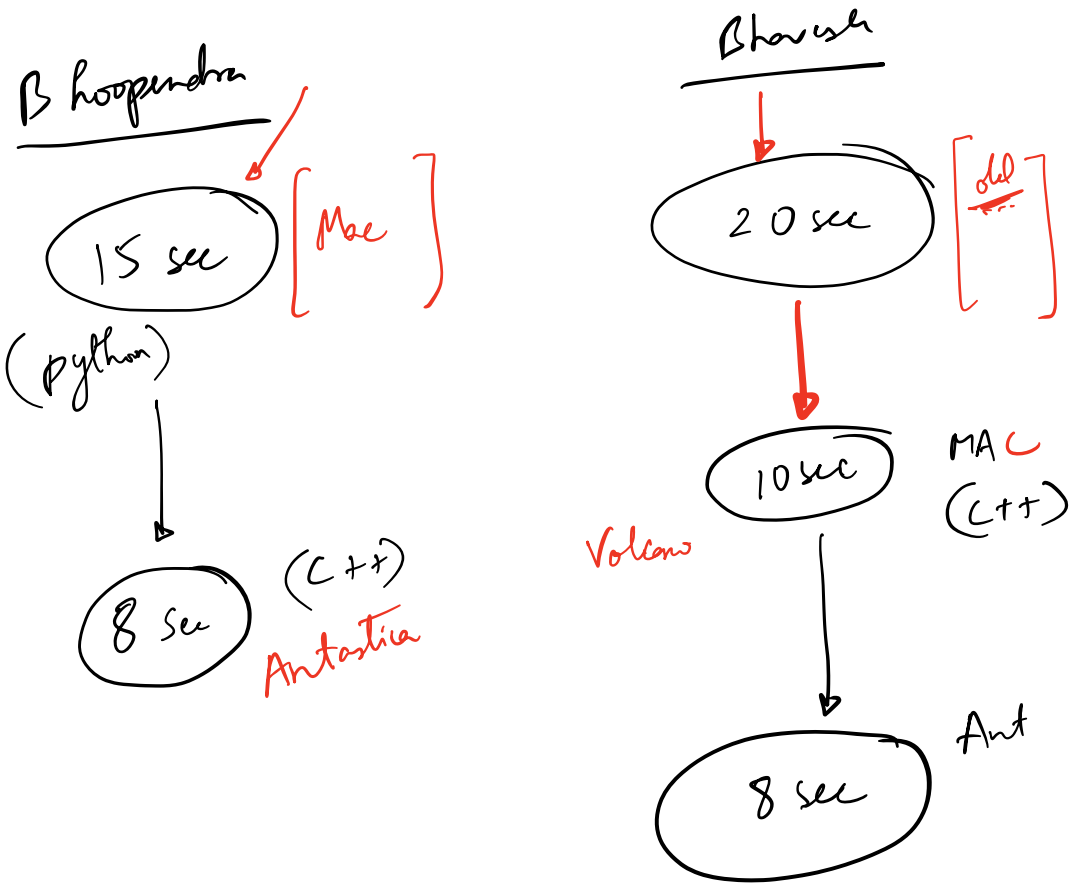
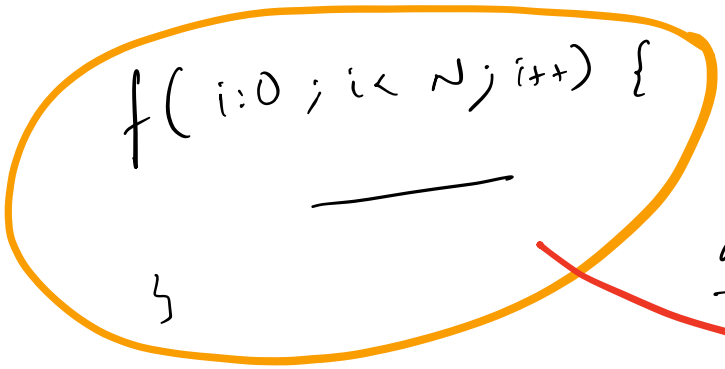
Time Complexity – 2

1) Given 10^4 no's, write an algo to sort those nos.

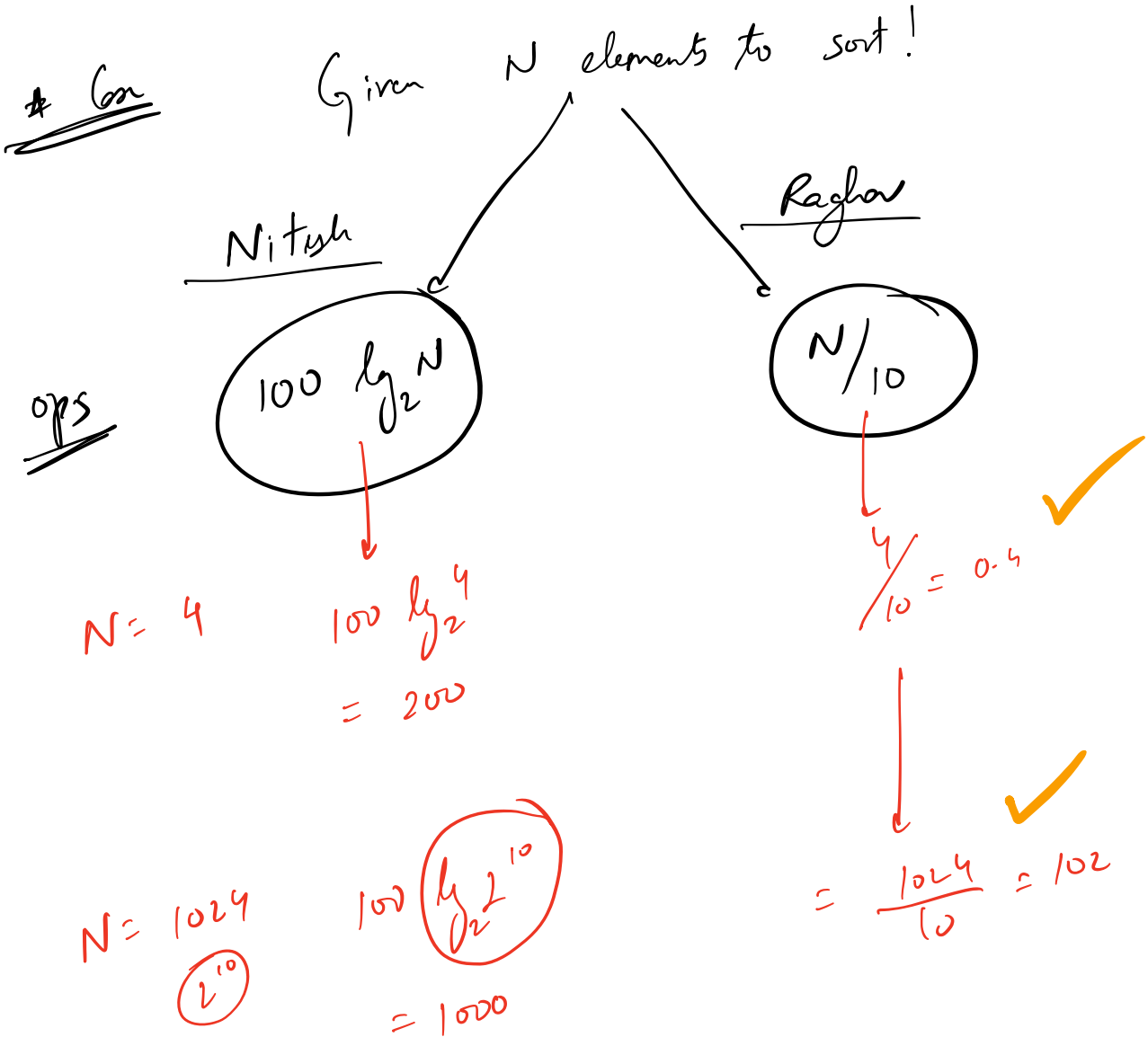


* Execution time is not a good factor
  - → S/w + H/w + outside factors



ops / iterations would be same for an algo!

N iterations ---



$N = 2^{20}$

$\sim 10^6$

$100 \log_2 2^{20}$ ✓

$100 \cdot 20$

$= 2000$

$\frac{10^6}{10} = 10^5$

100000

$\frac{10^9}{10} = 10^8$

$100\,000\,000$

$N = 2^{30}$

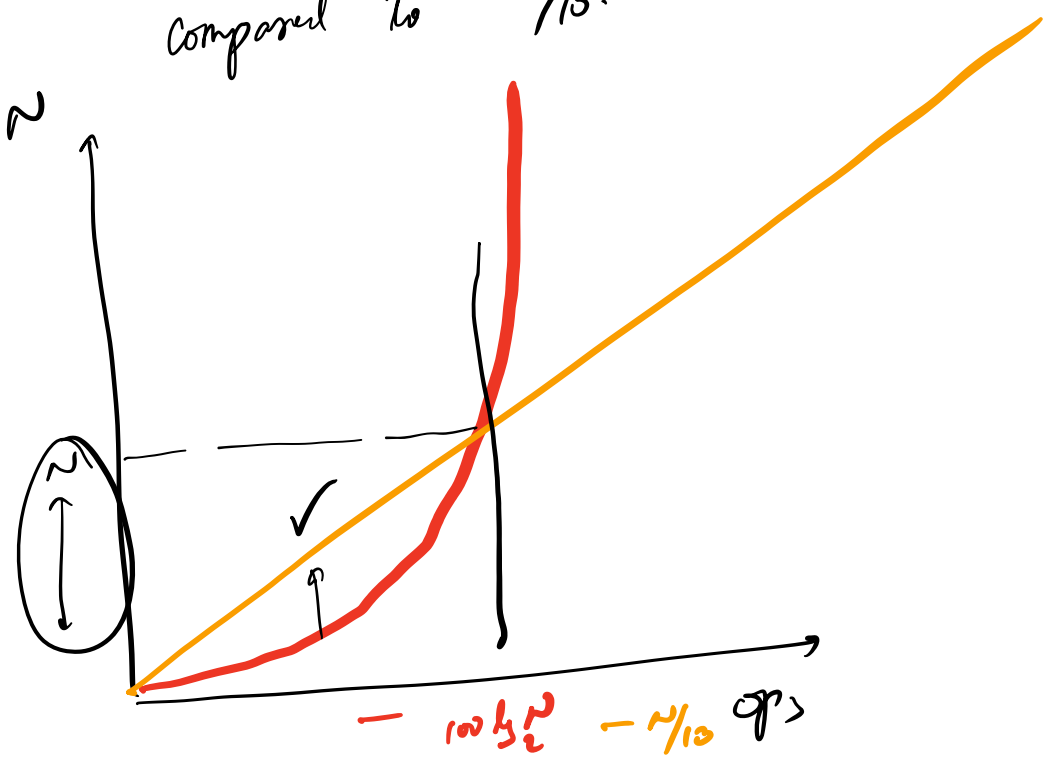
$\sim 10^9$

$100 \log_2 2^{30}$

$100 \cdot 30$

3000

* $100 \log N$ would perform worse for smaller values of N compared to $N/10$.



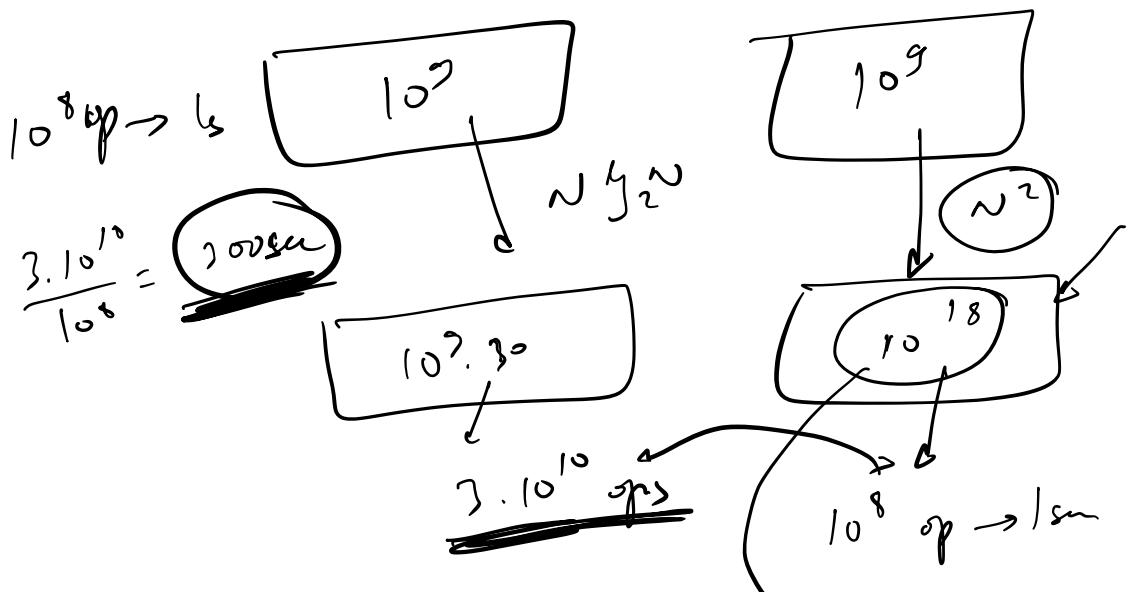
① Algo 1, Algo 2, Algo 3. . . Algo 5.

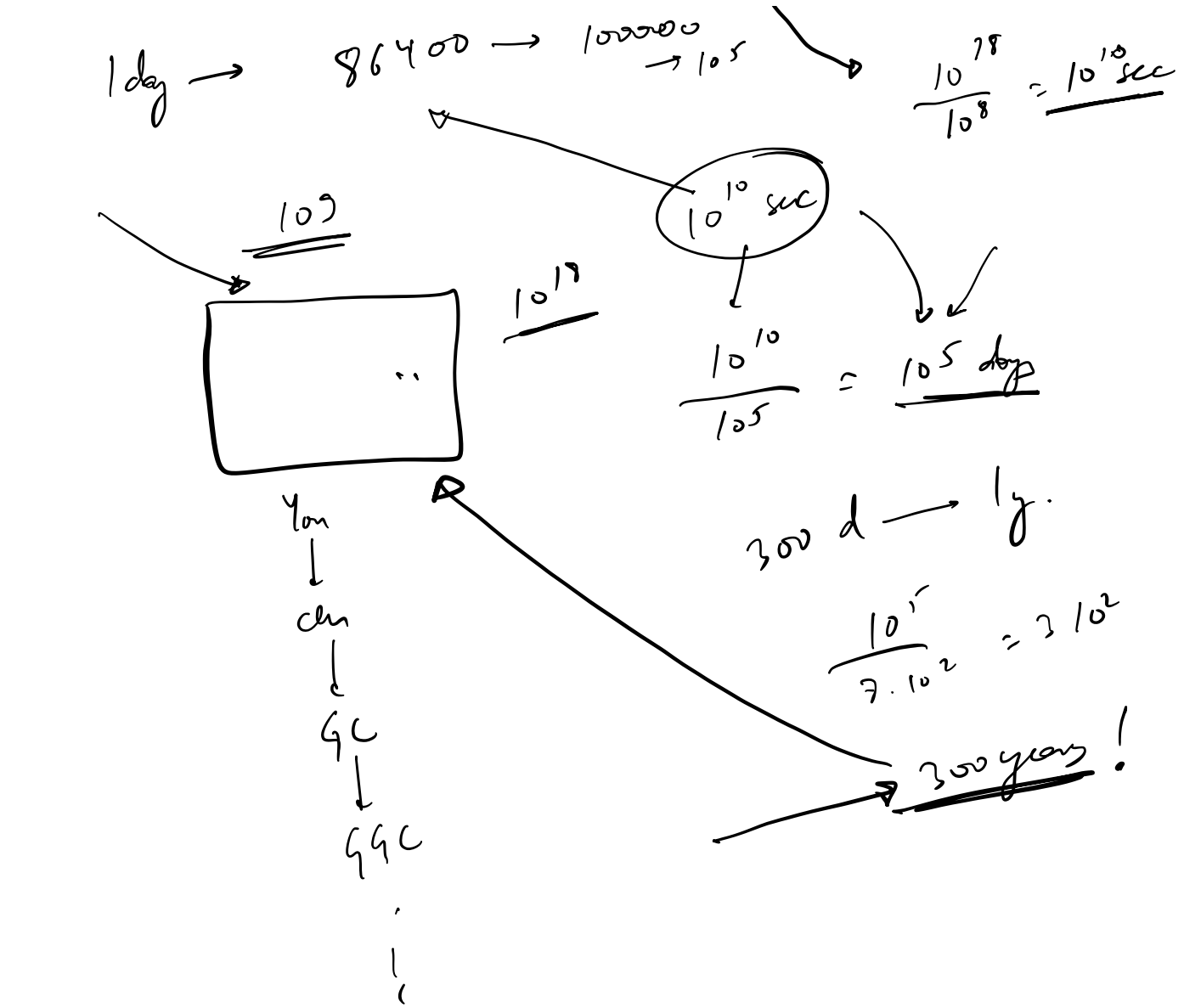
Input size is large for real applications!

Hotstar	Dmp	Google Search
3.3×10^7	$3 \cdot 10^9$	$5 \cdot 10^8$

Asymptotic Analysis of Algo's

↳ judge the performance of your algo for very large input values!





⊙ Big O Notation for judging algos.

1) Calculate the # ops / iter ...
2) Neglect lower order terms!
3) Neglect the constant coeff.

Srikant

$N \rightarrow$ Input

$N^2 + 10N$ op.

$N = 100$

$100^2 + 10 \cdot 100$

11000

$\frac{1000}{11000} \times 100\% = \sim 10\%$

$N = 10^5$

$(10^5)^2 + 10 \cdot 10^5$

$10^{10} + 10^6$

$10000\ 10 - -$

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

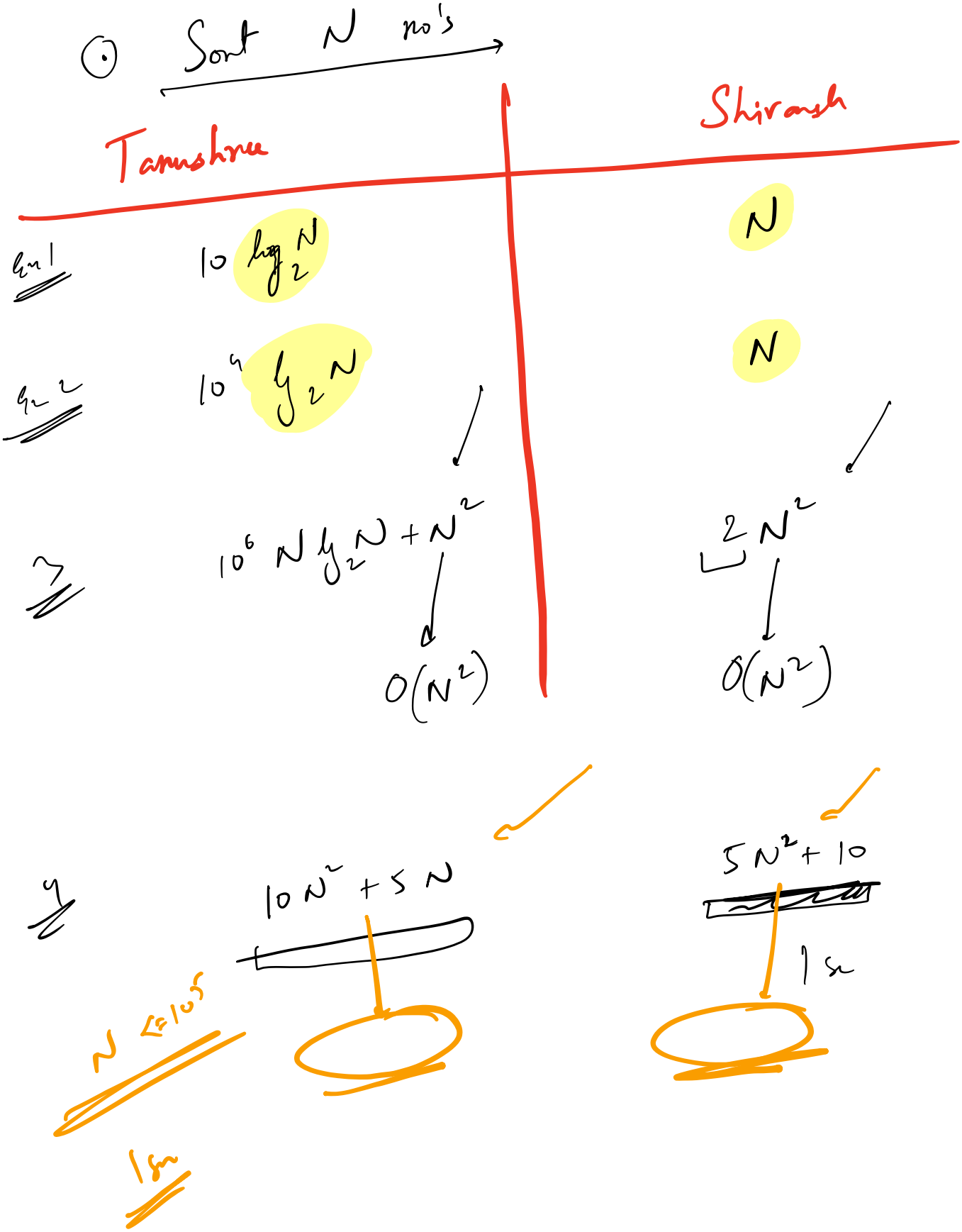
$= \frac{10^6}{10^{10}}$:-

$= 0.01\%$

$f(N) = 10N^2 + \underbrace{5N + 60 \cdot N^0}_{x}$

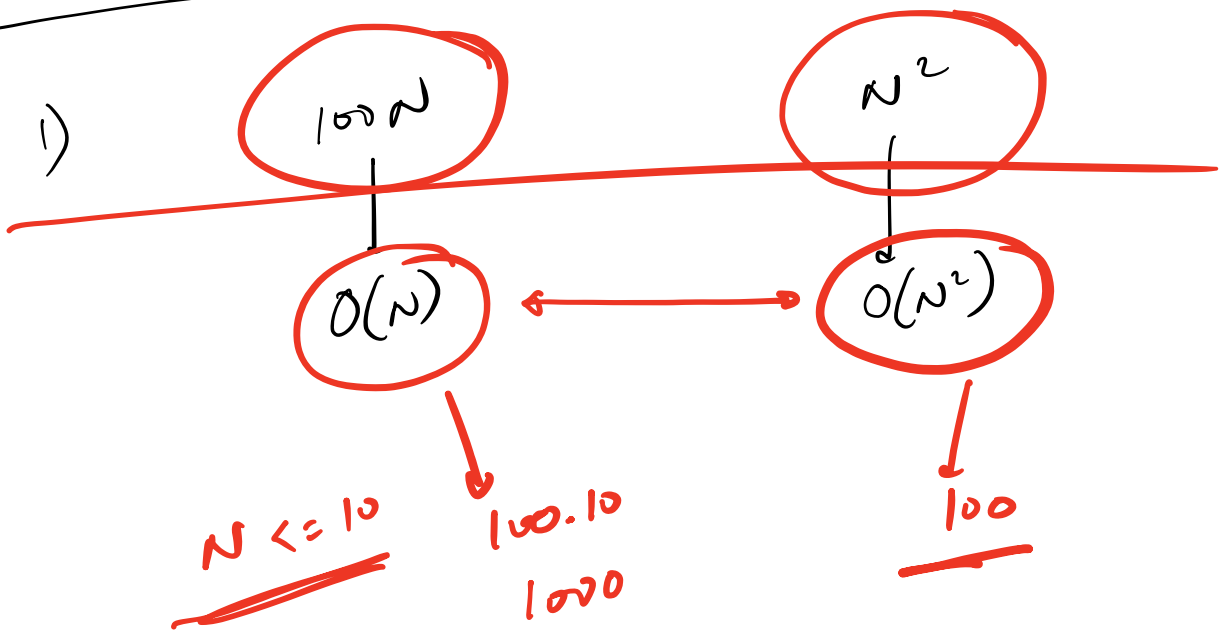
$10N^2$

TC: $O(N^2)$


Sort N no's
Tanushree
Shirash
N
N
$10 \log_2 N$
$10^4 \log_2 N$
$10^6 N \log_2 N + N^2$
$O(N^2)$
$2N^2$
$O(N^2)$
$10N^2 + 5N$
$5N^2 + 10$
N <= 10^5
1 sec
1 sec

Issues with Big O

1)



2

$10N^2 + N + 100$

$O(N^2)$

$5N^2 + 4$

$O(N^2)$

N, M

$f(N, M) = 5N \log_2 N + 10M^3$

$O(N \log N + M^3)$

$f(N, M) = 15M^2 + 5M$

$O(M^2)$

$f(N, M) = 15M^2 + NM + N^2$

$O(M^2 + NM + N^2)$

$O(M^2 + N^2)$

Space Complexity → Big O

int	4B
long	8B
double	8B

```
fun ( int N) {
    int x = N;        4
    int y = 10;       4
    long z = 100;     8
    double w = 10.0;  8
}
```

$f(N) = 24$

SC: O(1)

```
fun ( int N) {
    int n = N;        → 4 B
    — y = N;          → 4 B
    long z = 10;      → 8 B
    double w = 10.0;  → 8 B
    int arr[N];       → 4N B
}
```

$$f(N) = 24 + 4N$$

SC: $O(N)$

```
fun ( int N) {
    int n = N;        → 4 B
    — y = N;          → 4 B
    long z = 10;      → 8 B
    double w = 10.0;  → 8 B
    int arr[N];       → 4N B
    int b[N][N];      → 4N^2 B
}
```

$$f(N) = 24 + 4N + 4N^2$$

$O(N^2)$ SC

Q Given N no's - find their sum!

```
int findSum( int arr[], int N){
    int sum = 0;
    for(i: 0 ⟶ N){
        sum = sum + arr[i];
    }
    ret sum;
}
```

TC: O(N)

4B
4B

f(N) = 8

SC: O(1)

* Space Complexity → Amount of Extra Space taken by your algo other than the input

```
f(i:0 ⟶ N){
    int n = 10
}
```

4B X

① SC is the MAX amt of space taken by algo at any point in time!

```
func ( int arr[], int N, int K) {
    f ( i:0 ⟶ N ) {
        if (arr[i] == K)
            ret True;
    }
    ret False;
}
```

find K exists

SC

4B : O(1)

TC

Best Case : 1 op

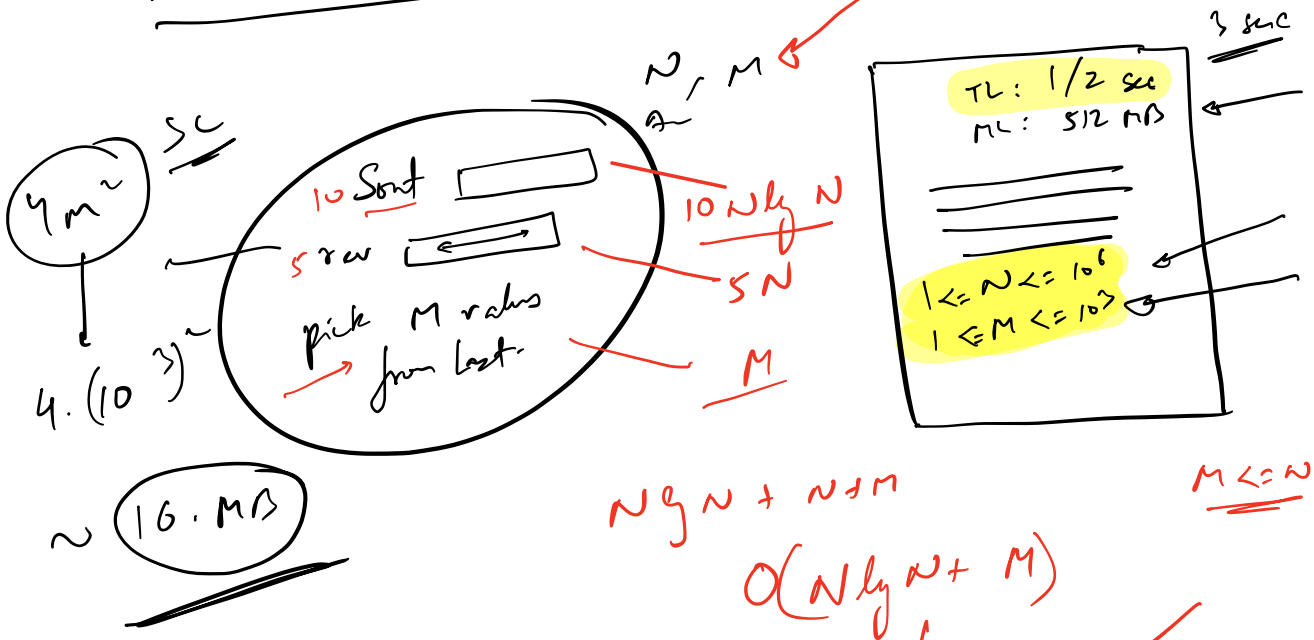
Worst Case : N op.

BC TC : O(1)

WC TC : O(N)

④ Big O deals with Worst Case TC. !

TLE : Time Limit Exceeded ! MLE !



N lg N + N + M

O(N lg N + M)

M <= N

$O(N \lg N)$

#op → $10 N \lg N + 5N + M$

#op → $10 \cdot 10^6 \cdot 20 + 5 \cdot 10^6 + 10^3$

$2 \cdot 10^8 + 5 \cdot 10^6 + 10^3$

$\sim 2 \cdot 10^8$ op

10^8 op → 1s

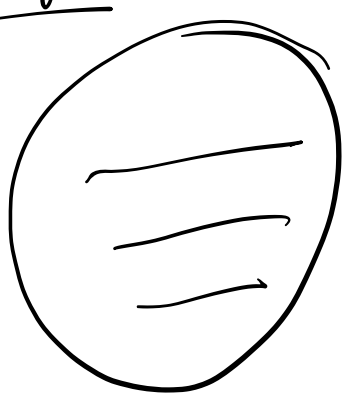
TL > ← ~ 2 sec !

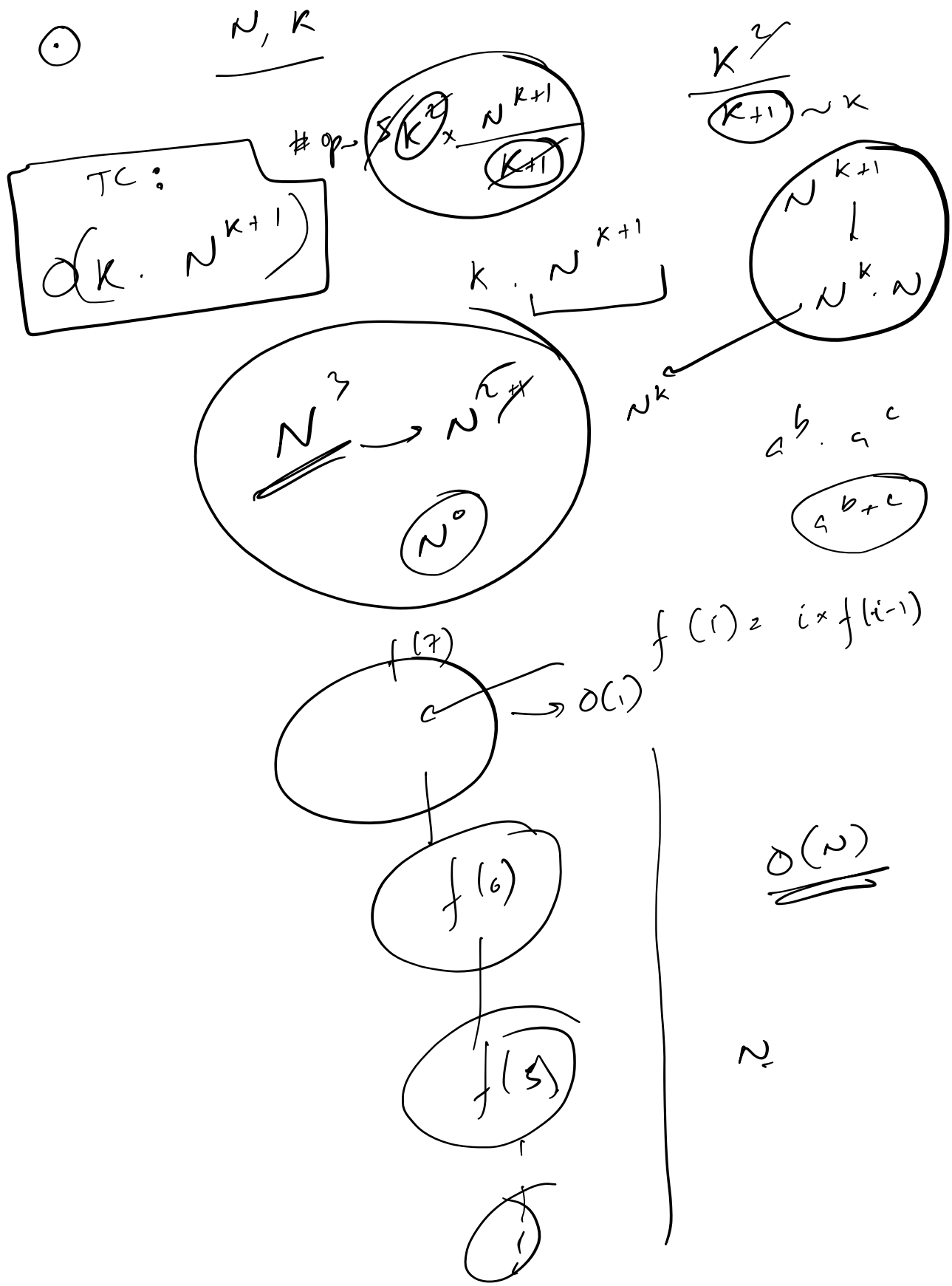
Algo →

TL: 1sn

5×10^8 op

X




N, K
TC:
O(K · N^{K+1})
# op
K^2 x N^{K+1}
K+1
K^2
K+1 ~ K
N^{K+1}
N^K · N
K · N^{K+1}
N^3 → N^{n-1}
N^0
N^K
a^b · a^c
a^{b+c}
f(7)
→ O(1)
f(1) = i × f(i-1)
f(6)
f(5)
1
O(N)
N