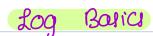
Time complexity
— Log Bouics
— comparing Iterations wing Graph
— Time complexity { Definition & Notations }
TLE
— Importance of contraints.



log (a) = c so that it equals a

To what power should I raise b

$$\Longrightarrow$$

$$\Rightarrow$$
  $b^{c} = a$ 

## Examples

 $\log_2(64) = 6$  :  $2^6 = 64$ 

 $log_3(27) = 3$ 

 $\log_{5}(25) = 2$   $\log_{2}(32) = 5$ 

$$\log_b(b^c) = c$$

 $^{\prime}\log_{2}(2^{6}) =$ 

 $\log_3(3^3) = 3$  $\log_5(5^2) = 2$  $\log_2(2^5) = 5$ 

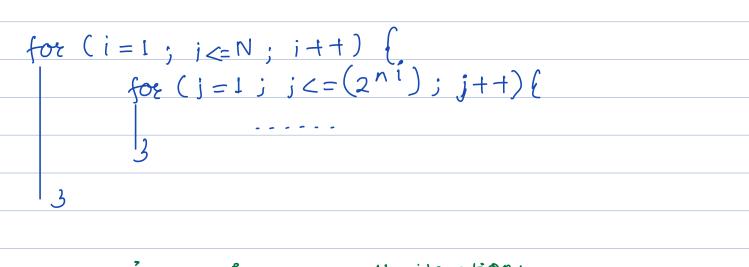
Q> Given a positive integer N, how many times do we need to divide by 2, until it neaches 1 Integer division  $\frac{5}{3} = 2$  $\frac{10}{2} \longrightarrow \frac{5}{2} \longrightarrow \frac{2}{2} \longrightarrow \frac{1}{3}$ N = 10 $\frac{9}{2} \longrightarrow \frac{4}{2} \longrightarrow \frac{2}{2} \longrightarrow \frac{1}{3}$ N = 9 $\frac{30}{2} \longrightarrow \frac{15}{2} \longrightarrow \frac{7}{2} \longrightarrow \frac{3}{2} \longrightarrow 1 \quad \mathsf{U}$ N = 30N = 27Given N, keep dividing by until we reach 1  $\frac{N}{2} \longrightarrow \frac{N}{11} \longrightarrow \frac{N}{2} \cdots 1$ Assume it will take k steps to seach 1  $\frac{N}{2} \longrightarrow \frac{N}{2^2} \longrightarrow \frac{N}{2^3} \longrightarrow \frac{N}{2^k}$  $\frac{N}{2^{K}} = 1 \implies N = 2^{K}$   $\log_{2} N = \log_{2} 2^{K}$ Only take int value or floor (log\_N)

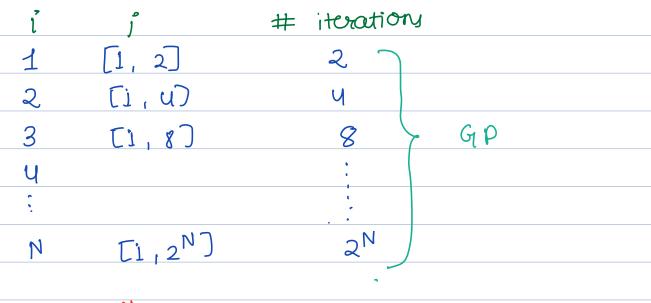
```
Floor of a value -> floor (1.5) -> 1
                                   (2.2) \longrightarrow 2
                                    hoor (2.99) -> 2
Storations
   i = N
   while (1>1) f
        i = i/2 # iterations
                                                     \rightarrow log (N)
              \frac{N \longrightarrow N}{2} \xrightarrow{N} \xrightarrow{2^2} \dots \qquad 1
                       log, N steps
N
               \frac{100}{2} \xrightarrow{50} \frac{50}{2} \xrightarrow{2} \frac{12}{2} \xrightarrow{6} \frac{3}{2} \xrightarrow{1}
 100
for (i = L j i < N j i = i * 2)
         \rightarrow 2 \longrightarrow 4 \longrightarrow 8 .....
                 after K steps we become or exceed N
                2^{k} = N
            log, 2^k = log N
                  t = log N
```

```
for (i = 0; i < N; i = i*2)
                          0 \longrightarrow 0*2 \longrightarrow 0*2^2 \cdots
step 1
                      \# iterationy \longrightarrow \infty
stop 2
step 3
for (i=1; i<=10; i++) {
        for (j=1; j<=N; j++ {
  3
                         no of iterations
           [1, N]
                           N
          [N, 1]
              [1, N]
        10
                            10* N
```

```
for (i=1; i <= N; i++) {
        for (j=1; j<=N; j++ {
                      no of iterations
           [1, N]
          [N,1]
                         N
                               Ntimes
          [1, N]
           # iteration = N* N
for (i=1; i <= N; i++) {
      for (j=1; j \leq N; j=j*2)
              j no of iterations
 .3
         log_N log_N
         log<sub>2</sub> N log<sub>2</sub> N
                               Ntimes
             log, N log, N
```

# iteration = N\* log N

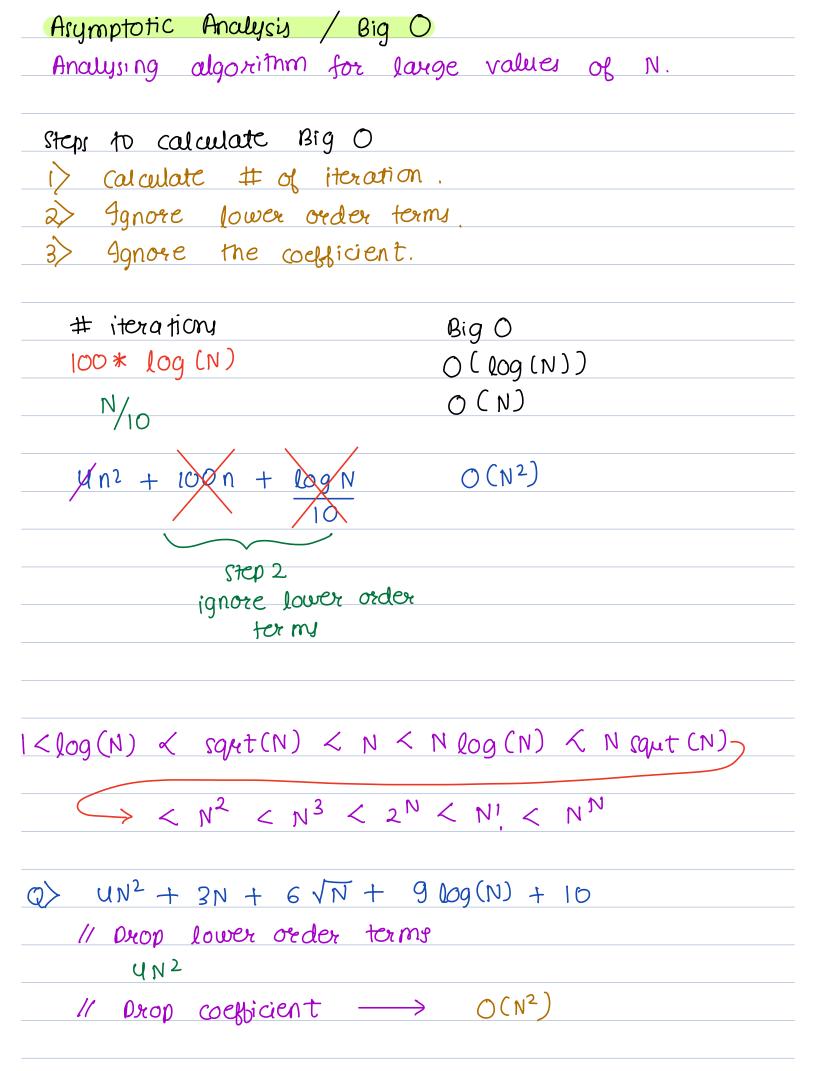




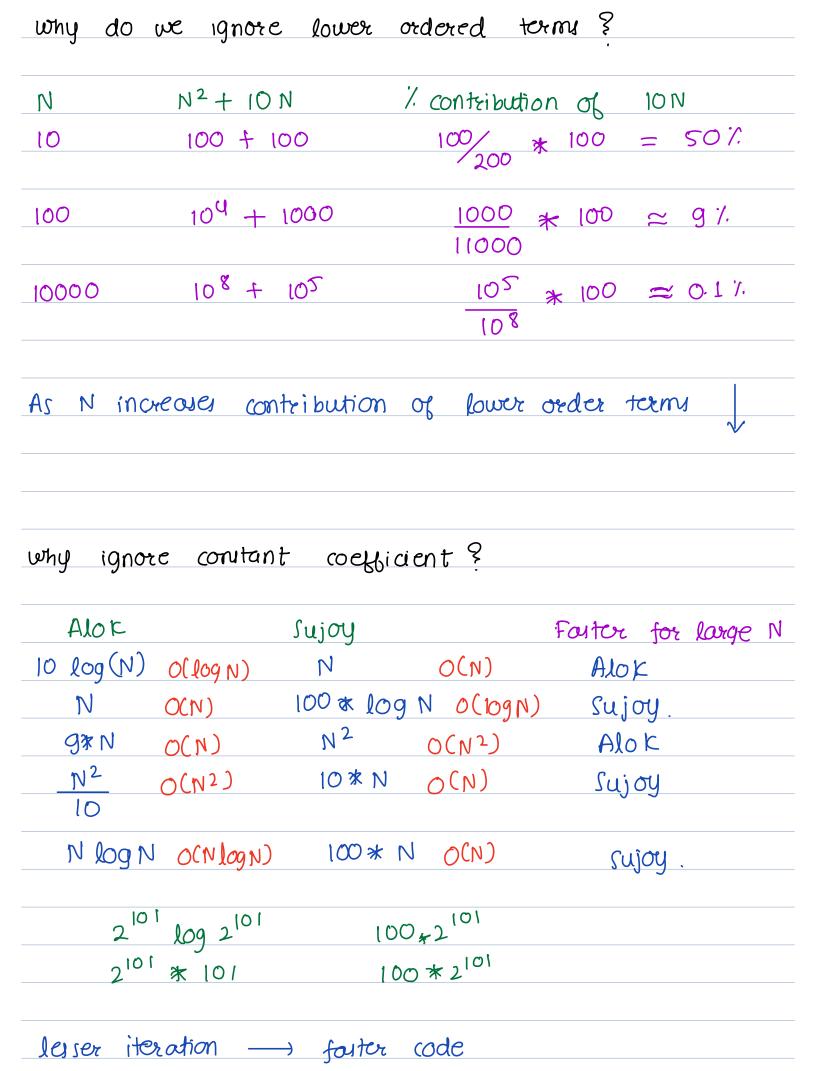
$$\frac{2*(2^{N}-1)}{2^{-1}} = 2*(2^{N}-1)$$

Comparing Ateration

	Sumit	Jahnavi
	Algo 1	Algo 2
rations>	100* log (N)	N/10
N <= 3500	Jan navi 3 (	Mgo2 was fostere
N > 3200	sumit's	ulgo 1 was fastere
al world		
world cup —	—→ S.6 ¢	r
youtube -	10 + B	
9030,333		
real world the	volue of N	is huge
		Break: 22:38
		13MW . 22.50



Q >	NV	+ 3	N log N	+ 1				
			NlogN					
			CNLOGN.					
			J					
Q>	UN	log N	+ 3 N	1/N +	106			
			higher					
					NIN		GU	
					64*8	]		
					higher or			
sten	2 —	$\rightarrow$	3 N 1 N					
ster	) 3 —		0 (NV	N)				
				•				



Coeks	doesn't	matter	for	Big	0	
0			_			

## Limitation of Big O Notation

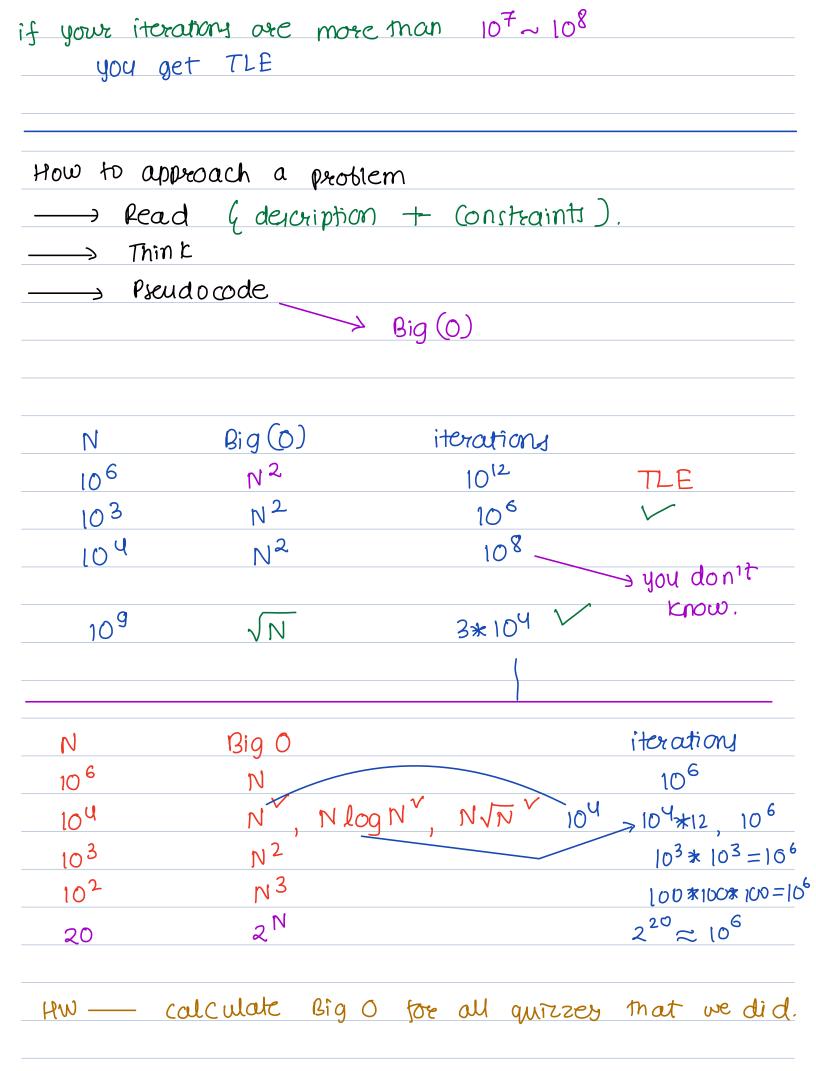
Algo 1	Algo 2
n <sup>2</sup> + 10 n	$n^2 + 100n$
	10
O(n <sup>2</sup> )	O(n2)

Both algor are same

If Big O itself 4 same, compare the no. of iterations

2> Big	0 only works for	very large v	ralues. of N
	Algo L	Algo 2	
N	1000 N <sup>2</sup>	N3	
10	105	1000	Algo 2
100	1000* 104	10 <sup>6</sup>	Algo 2
1000	1000 * 106	109	same
10000	1000 * 108	1012	Algo L.

Time limit Exceeded submit TLE read - Think - pseudocode - Code Online editor - The server runs on 1GHz... 109 instructions / second. No. of intention / iteration ~ 10 to 100 for for : 3 10 instruction : 3 100 instruction iteration Time limit to reun your code I second.  $1 \longrightarrow 10^9$  instructions small  $1 \longrightarrow 10^8$  iterations  $\longrightarrow$  10<sup>9</sup> instructions ) longe  $1 \longrightarrow 10^{7}$  iterations 107~108



Anstruction — smallest calculation or step a CPU will do.

for 
$$(i = 1)$$
;  $(i < 1)$ ;  $(i = i + 1)$ )  $($ 

1 iteration -> 10 instructions 3 small code 100 instruction 3 large code

 $n + 10^6$   $n^2$ 

\_\_\_\_\_i 1 to N

f(\*) = = N return

i == VN return

IN iteration