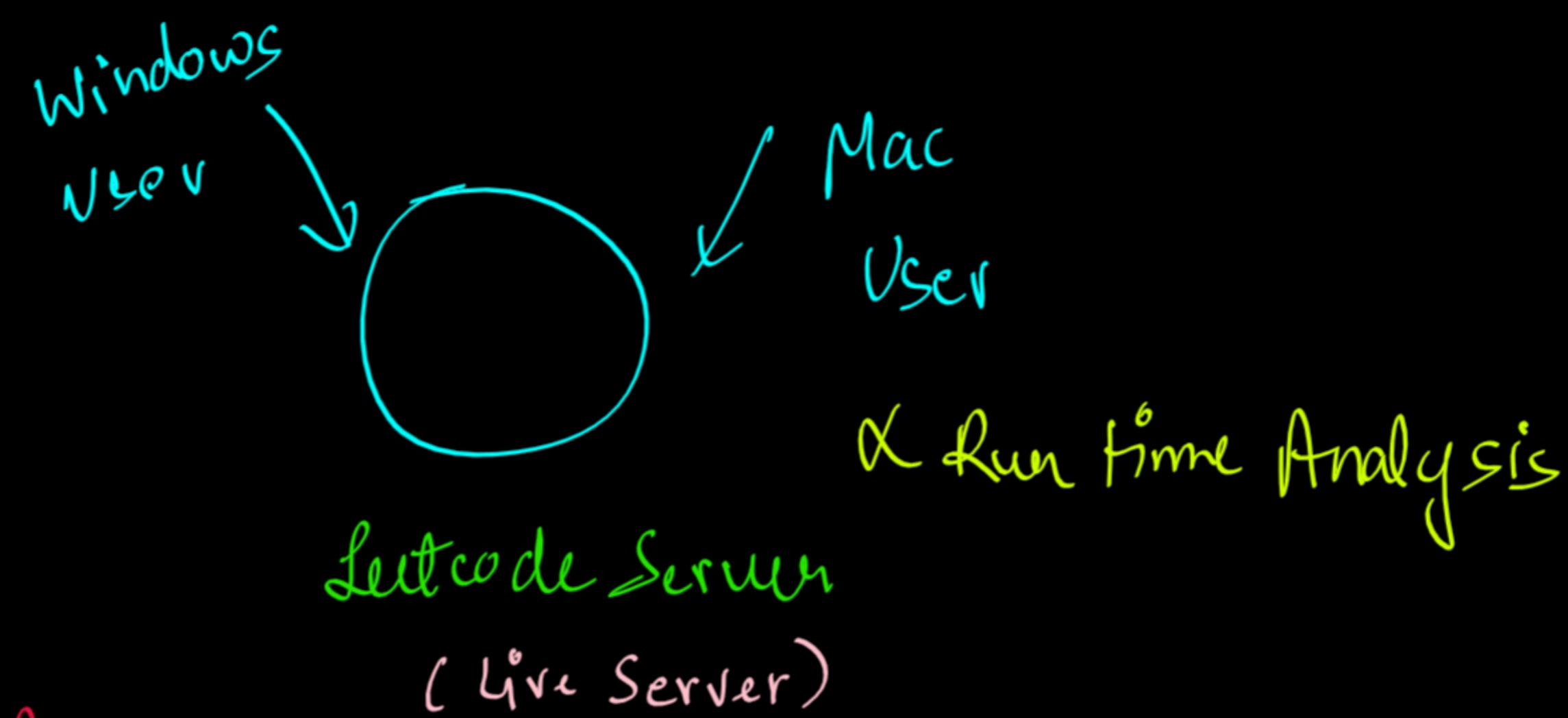


# Time and Space Complexity

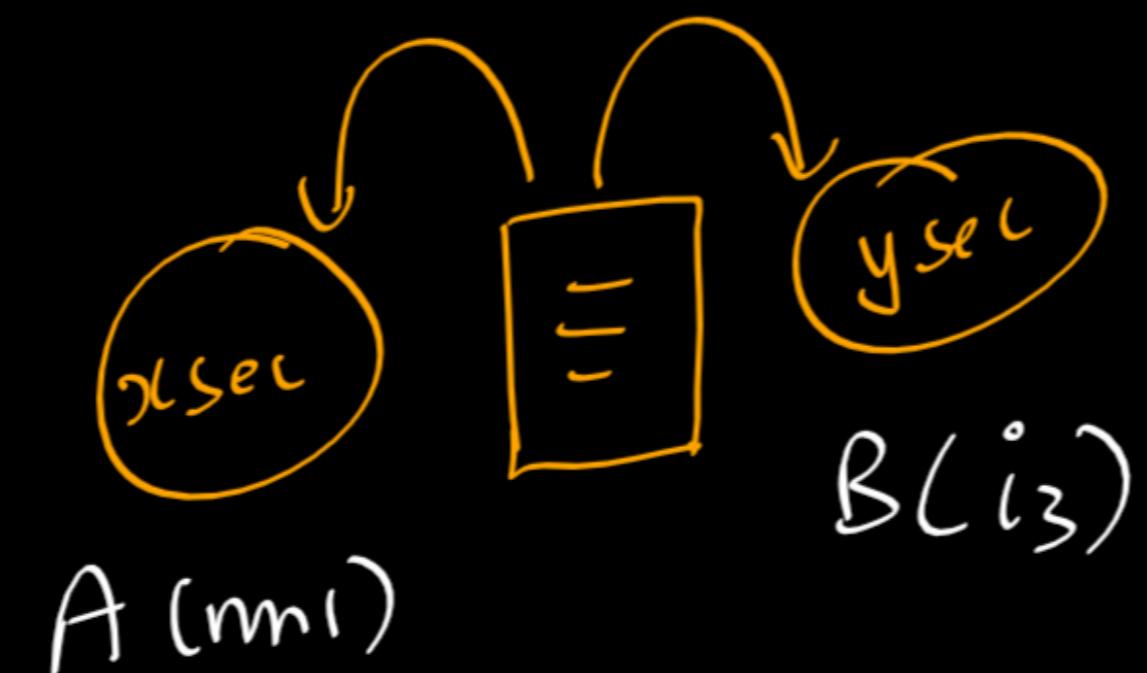
{ 200ms, 250ms, 1ms, 1000ms } → Run-time (Machine Dependent)



## Asymptotic Analysis

function of Runtime

$T(n)$  dependent on input size

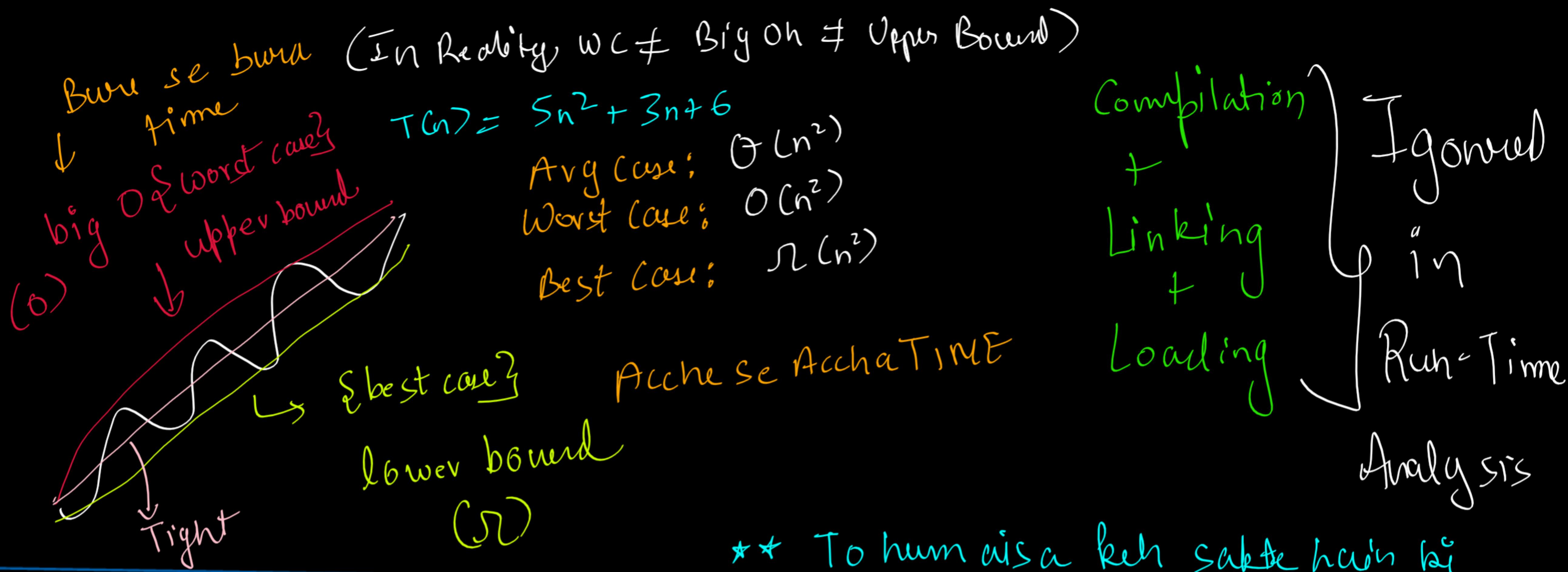


Algorithm  $\alpha$  <sup>input size</sup> ↗ growth of Runtime wrt input size.  
ignore rest all the factors.

↓  
machine independent

for A

input  $N \geq 100$  RunTime  $\leq 100\text{ms}$  } Linear  
 $N = 1000$  RunTime  $\leq 1000\text{ms}$  } Dependence



bound  
(Avg Case)  $\Theta$

\*\* To hum cisa keh sakte hain ki  
jab program process ban gaya hai  
us k baad ka tym run-time hogा

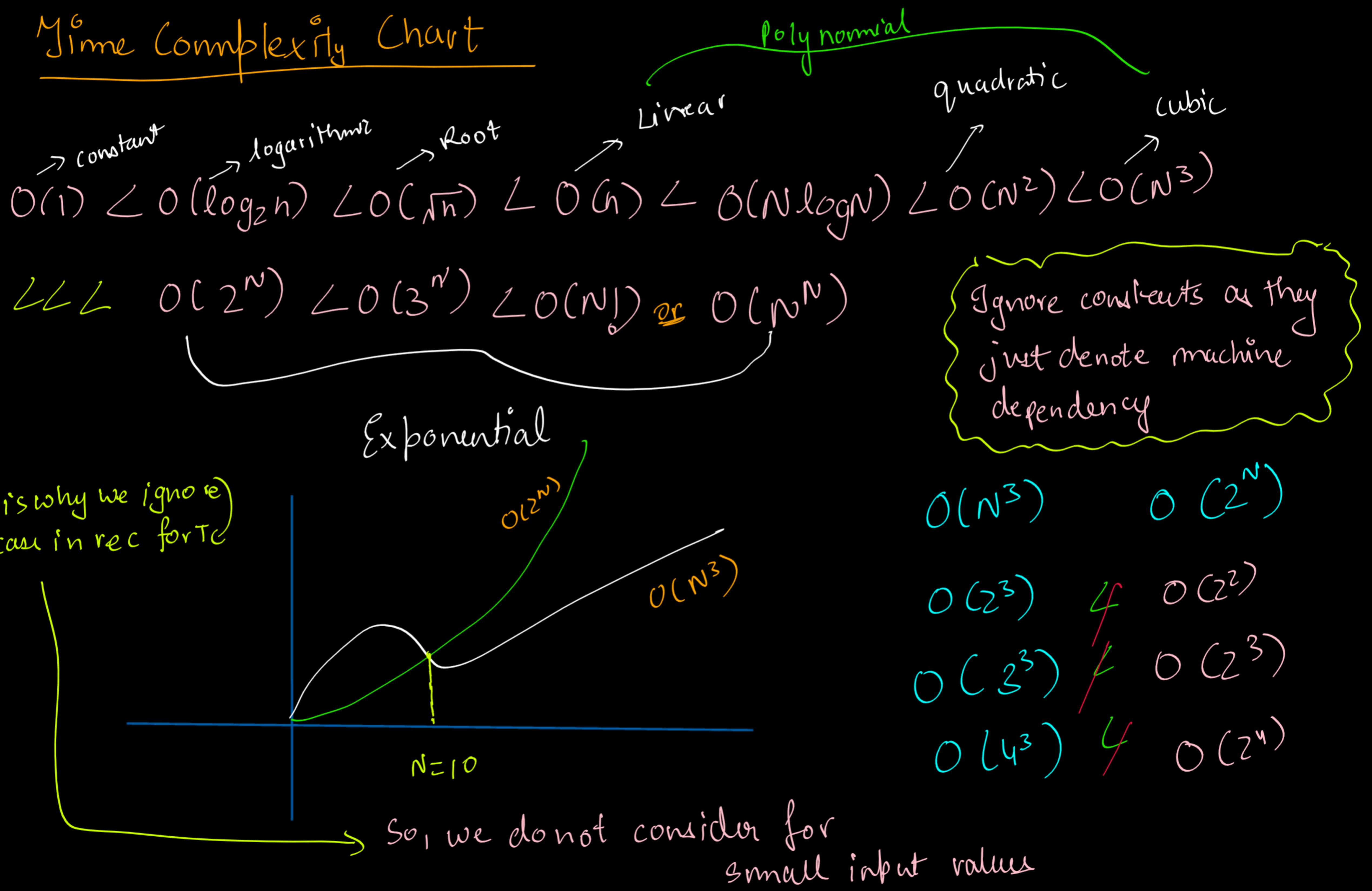
\*\* Best analysis is WORST CASE Scenario.

Bure se bure din me koi algo kaise perform Karti hui wahi  
comparison best rahega.

Yani jab program RAM se AAJATA HAI.

## Amortized Poochna Hai (during own vector design)

### Time Complexity Chart



## for loops Time Complexity

```
for (int i = 0; i < n; i++) {  
    for (int j = 0; j < n; j++) {  
        System.out.println(i + " " + j);  
    }  
}
```

$n=5$

Using Property

$$f(n) * O(g(n)) \\ = O(f(n) * g(n))$$

$$i=0 \quad j = 0, 1, 2, 3, 4 \quad (5k) = O(n)$$

$$i=1 \quad j = 0, 1, 2, 3, 4 \quad (5k) = O(n)$$

$$i=2 \quad j = 0, 1, 2, 3, 4 \quad (5k) = O(n)$$

$$i=3 \quad j = 0, 1, 2, 3, 4 \quad (5k) = O(n)$$

$$i=4 \quad j = 0, 1, 2, 3, 4 \quad (5k) = O(n)$$

$$n * O(n) \\ = \boxed{O(n^2)}$$

```

for (int i = 0; i < n; i++) {
    for (int j = n - 1; j >= 0; j--) {
        }
    }
}

```

$$n=5$$

$i=0$	$j = 4, 3, 2, 1, 0$	$k+k+k+k+k \rightarrow O(n)$	$n * O(n)$
$i=1$	$j = 4, 3, 2, 1, 0$	$k+k+k+k+k \rightarrow O(n)$	
$i=2$	$j = 4, 3, 2, 1, 0$	$k+k+k+k+k \rightarrow O(n)$	
$i=3$	$j = 4, 3, 2, 1, 0$	$k+k+k+k+k \rightarrow O(n)$	
$i=4$	$j = 4, 3, 2, 1, 0$	$k+k+k+k+k \rightarrow O(n)$	

$$= O(n^2)$$

```

for (int i = 0; i < n; i++) {
    for (int j = 0; j <= i; j++) {
    }
}

```

\*4 Humsha

nested-for

$O(n^2)$

$$k \left( \frac{n(n+1)}{2} \right)$$

Nahi  
Hota

i=0      j= 0 → k

i=1      j= 0,1 → 2k

i=2      j= 0,1,2 → 3k

i=3      j= 0,1,2,3 → 4k

i=4      j= 0,1,2,3,4 → 5k

$O(n^2)$

```
for (int i = 1; i < n; i *= 2) {  
}
```

$$n = 35$$

$$i = 1 \rightarrow 1$$

$$i = 2 \rightarrow 2$$

$$i = 4 \rightarrow 3$$

$$i = 8 \rightarrow 4$$

$$i = 16 \rightarrow 5$$

$$i = 32 \rightarrow 6$$

$$O(\log_2 35) \approx O(\log_2 n + 1)$$

iterations

$$O(\log_2 n)$$

```

for (int i = 0; i * i < n; i++) {
}

```

$$n = 64$$

(b)  $i = 0 \leq 64$

(b)  $i = 1 \leq 64$

for  $64 \rightarrow$  runs for 8 times

(b)  $i = 2 \leq 64$

$\sqrt{n}$  times

(b)  $i = 3 \leq 64$

$$O(\sqrt{n})$$

.

(b)  $i = 4 \leq 64$

(b)  $i = 5 \leq 64$

$$O(\sqrt{n+1}) \approx k$$

(b)  $i = 6 \leq 64$

(b)  $i = 7 \leq 64$

$$\approx O(\sqrt{n})$$

(b)  $i = 8 \leq 64$



## # Algorithms - Competitive Programming Chart!

Input Size ( $n$ )	Time Complexity (Worst case)	Hint of Algorithm
$\leq 10$	$O(n!)$ or $O(2^n)$	Backtracking [e.g. permutation, subsets]
$\leq 18$	$O(2^n \cdot n)$	Travelling Salesman
$\leq 22$	$O(2^n \cdot n^2)$	DP with Bitmasking
$\leq 10^2$	$O(n^4)$	Quadruplets / 4 Nested loops 
$\leq 4 \times 10^2$	$O(n^3)$	3 nested loops, Floyd-Warshall, 
$\leq 2 \times 10^3$	$O(n^2 \log n)$	Nested loops & Binary Search
$\leq 10^4$	$O(n^2)$	2 Nested loops (Bubble, Insertion, Selection) 
$\leq 10^5 - 10^6$	$O(n \log n)$	Sorting (QS, MS), BS in Answer, Greedy, etc.
$\leq 10^8$	$O(N)$	Array & Strings,
$\leq 10^{18}$	$O(\log N)$ or $O(1)$	mathematical formula

Leetcode  
Codechef  
Codeforces

$\rightarrow 1s (10^8$  operations)

for ex: if  $N$  is of the order  $10^8$   
then max complex will be  $O(N)$   
as  $10^8$  max op will be performed in 1s.