

Time Complexity

Big O notation - upper bound

Theta - avg bound

Omega - lower bound

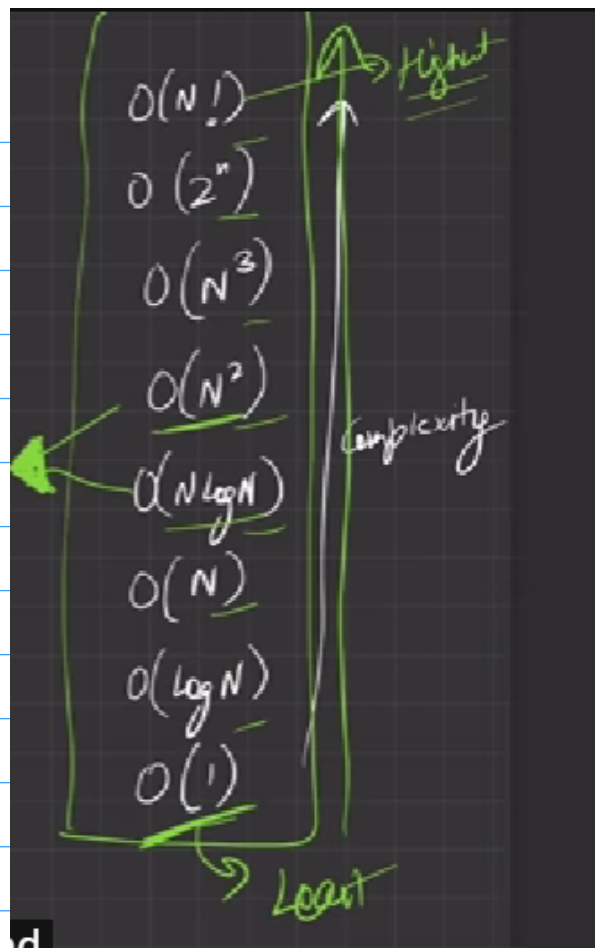
```
for (int i=0;i<10;i++){  
    cout<<"Hello";  
}
```

<-- This will take
time complexity of
 $O(1)$ i.e constant time

```
for (int i=0;i<n;i++){  
    cout<<"hello"  
}
```

<-- this will take a time
complexity of $O(n)$ i.e
linear time

logarithmic time = $O(\log n)$	generally in binary search
Quadratic time = $O(n^2)$	generally in double for loop
Cubic time = $O(n^3)$	generally in triple for loop



$$f(n) \Rightarrow 5n^2 + \log n \rightarrow O(n^2)$$

always ignore constant and lower time complexity

What is the time, space complexity of following code :

```
int a = 0, b = 0;
for (i = 0; i < N; i++) {
    a = a + rand();
}
for (j = 0; j < M; j++) {
    b = b + rand();
}
```

Assume that rand() is $O(1)$ time, $O(1)$ space function.

here two independent 'for' loops are going on ,so time complexity will get add.

therefore time complexity is $O(n+m)$

if nested loop is there then $O(n*m)$

The image shows handwritten notes on a chalkboard, organized into two main columns. The left column is titled 'Constraints' and lists various ranges for n . The right column is titled 'Time Complexity' and lists several common complexity classes. A large bracket on the right side of the complexity list is marked with an 'X' and the text 'TLE' (Time Limit Exceeded), indicating that those complexities are inefficient for the given constraints.

Constraints

- $1 < n < 10^6$
- $1 < n < 1000$

Time Complexity

- $\leq [10..11]$
- $< [15..18]$
- < 100
- < 400
- < 2000
- $< 10^4$
- $< 10^6$
- $< 10^8$

Time Complexity List:

- $O(n!)$, $O(n^!)$
- $O(2^n * n^2)$
- $O(n^4)$
- $O(n^3)$
- $O(n^2 * \log n)$
- $O(n^2)$
- $O(n \log n)$
- $O(n)$, $O(\log n)$

A large bracket on the right side of the complexity list is marked with an 'X' and the text 'TLE'.