DSA - 90 Days

Day 1-2

Time Complexity - It is the amount of time taken by an algorithm to run.
- as a function of length of the input

Why ? - for making better programs
- Comparision of algo

- Big Oh notation → Upper Bound
- Theata Θ → for average Case Complexity
- Omega → Lower Bound

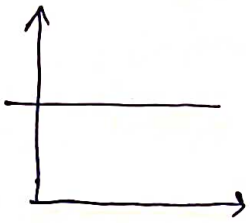
- **Constant Time** - $O(1)$
- **Linear Time** → $O(n)$
- **Logarithmic Time** → $O(\log n)$
- **Quadratic Time** → $O(n^2)$
- **Cubic Time** → $O(n^3)$

→ **Stuck in TLE** TLE → Timing limit exceeded

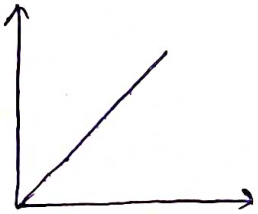
10^8 operation rule → Most of the modern machine can perform 10^8 operation/second.

Constraints	Time Complexity
$\leq [10 \ldots 11]$	$O(n!), O(n^6)$
$< [15 \ldots 18]$	$O(2^n * n^2)$
< 100	$O(n^4)$
< 400	$O(n^3)$
< 2000	$O(n^2 * \log n)$
$< 10^4$	$O(n^2)$
$< 10^6$	$O(n * \log n)$
$< 10^8$	$O(n), O(\log n)$

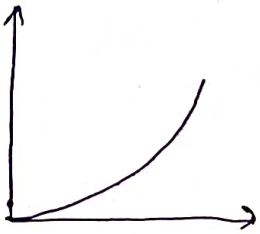
→ O(1)



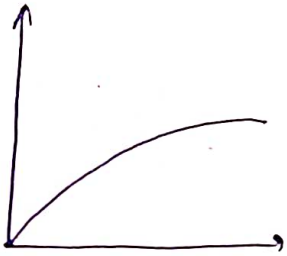
→ O(n)



→ $O(n^2)$



→ O(log n)



Big Oh Notation

O(N!) → Highest

O(2^n)

O(n^3)

O(n^2)

O(n log n)

O(n)

O(log n)

O(1) → least

Complexity