

What is 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 progress.
→ Comparison of Algo.

Big O notation → Theta Θ → Omega Ω
 ↓ ↓ ↓
 Upper bound for avg. case complexity Lower bound.

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

→ $O(1)$ Big O notation.

→ $O(n)$

→ $O(n^2)$

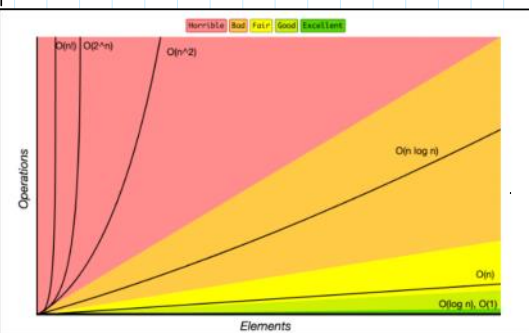
→ $O(\log n)$

Highest.

$O(n!)$
 $O(2^n)$
 $O(n^3)$
 $O(n^2)$
 $O(n \log n)$
 $O(n)$
 $O(\log n)$
 $O(1)$

Least

Complexity



How to Solve TLE Problem

→ There is a 10^8 rule →
 Which says that Most of the modern machine can perform 10^8 operation/second.

* There is an amazing table for Time Complexity:

Some Sample questions to calculate time Complexity.

- 1) $f(n) \rightarrow 2n^2 + 3n \rightarrow O(n^2)$
- 2) $f(n) \rightarrow 4n^4 + 3n^3 \rightarrow O(n^4)$
- 3) $f(n) \rightarrow N^2 + \log N \rightarrow O(n^2)$
- 4) $f(n) \rightarrow 12001 \rightarrow O(1)$
- 5) $f(n) \rightarrow 3n^3 + 2n^2 - 5 \rightarrow O(n^3)$
- 6) $f(n) \rightarrow 5n^2 + \log n \rightarrow O(n^2)$
- 7) $f(n) \rightarrow \frac{1}{4} \rightarrow O(1)$

While calculating time complexity we should keep in mind that we have to consider maximum time complexity only & ignore the smaller one always.
 ignore lowest degree & constants

Input Length	Worst Accepted Time Complexity	Usually type of solutions
10 - 12	$O(N!)$	Recursive and backtracking
15 - 18	$O(2^N * N)$	Recursion, backtracking, and bit manipulation
18 - 22	$O(2^N * N)$	Recursion, backtracking, and bit manipulation
30 - 40	$O(2^{N/2} * N)$	Meet in the middle, Divide and Conquer
100	$O(N^4)$	Dynamic programming, Constructive
400	$O(N^3)$	Dynamic programming, Constructive
2K	$O(N^2 * \log N)$	Dynamic programming, Binary Search, Sorting, Divide and Conquer
10K	$O(N^2)$	Dynamic programming, Graph, Trees, Constructive
1M	$O(N * \log N)$	Sorting, Binary Search, Divide and Conquer
100M	$O(N), O(\log N), O(1)$	Constructive, Mathematical, Greedy Algorithms