

**Your Ultimate Guide To Landing
Top AI roles**



Time Limit Exceed (TLE)



- While doing CP, Leetcode practice or during online assesment of Companies → we would definitely have encountered TLE Error.
- This happens when our program is not optimal and it can't complete execution for a given input size within some fixed time (say 1 second).
- A naive/bruteforce solution is almost never accepted. So how to know the expected time complexity?
- The answer is related to the no of operations that are allowed to perform within a second.
- Most of the platform these days allows 10^8 operation/sec.
- So Accordingly search for right complexity by looking at constraints in the problem.

Analysis: Time Limit $\rightarrow 1 \text{ sec}$

$1 \leq \text{Input size}(n) \leq 10^5 \leftarrow \text{Input Constraints}$



And Time Complexity of our Implemented Algorithm $= O(n^2)$

Will it give TLE?

\rightarrow If $n = 10^5$ (worst case)

maximum no of operations required to solve the problem $= O(n^2) = O((10^5)^2) = O(10^{10}) = c \cdot 10^{10}$

\hookrightarrow But we can do a maximum 10^8 operations. so it will give **TLE**

\rightarrow If our algorithm Time Complexity $= O(n \log n)$

maximum no of operations required to solve the problem $= O(n \log n) = O(10^5 \cdot \log 10^5) = c \cdot 10^5 \cdot 5 \cdot \log 10^4$

$$= c \cdot 5 \cdot 4 \cdot 10^5$$

$$= 2c \cdot 10^6 = c' \cdot 10^6$$

Pass \rightarrow NO TLE

Choose your algorithm such that, for largest input size your algorithm should perform no more than 10^8 operations. else it will give TLE

Constraints	Worst Time Complexity
$n \leq 12$	$O(n!)$
$n \leq 25$	$O(2^n)$
$n \leq 100$	$O(n^4)$
$n \leq 500$	$O(n^3)$
$n \leq 10^4$	$O(n^2)$
$n \leq 10^6$	$O(n \cdot \log n)$
$n \leq 10^8$	$O(n)$
$n > 10^8$	$O(\log n) / O(1)$

Some Notes for Python



→ Python is slower than C++/Java.

→ So same algorithm may pass in C++/Java but fail in Python if it is too close to limit.

→ So keep that into consideration while choosing algorithms.

→ For example on Leetcode

↳ in C++ we can do $\sim 10^8$ operations in 1 sec.

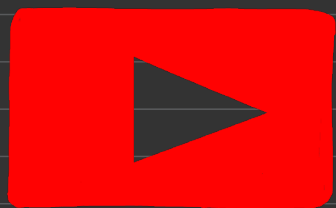
↳ but in Java, we can do $\sim 10^7$ operations in 1 sec.

→ so, for example

① $n \leq 10^7 \rightarrow O(n)$ algorithm.

② $n \leq 10^8 \rightarrow O(\log n)$ algorithm.

Like



Subscribe