

What is Time Complexity ?

Time complexity is a way to describe how the running time of an algorithm increases with the size of the input. It is often expressed using big O notation, which describes the upper bound of the running time in terms of the size of the input.





There are several types of Time Complexity :

- Constant time complexity - O(1)
- Logarithmic time complexity - O(log n)
- Linear time complexity - O(n)
- Quadratic time complexity - O(n^2)
- Exponential time complexity - O(2^n) o





- **Constant time complexity $O(1)$ -**

The running time of the algorithm remains the same, regardless of the size of the input.

- **Logarithmic time complexity $O(\log n)$ -**

The running time of the algorithm increases logarithmically with the size of the input. Algorithms with logarithmic time complexity are very efficient.



SWIPE



- **Linear time complexity $O(n)$** -
The running time of the algorithm increases linearly with the size of the input.
- **Logarithmic time complexity $O(\log n)$** -
The running time of the algorithm increases logarithmically with the size of the input.
- **Exponential time complexity $O(2^n)$** -
The running time of the algorithm increases exponentially with the size of the input.



SWIPE



AlgoTutor

BRUSH YOUR KNOWLEDGE
WITH EXPERTS
ON
TIME COMPLEXITY

14 - 01 - 2023

REGISTER NOW

Click for registration

