Bing

Hi, this is Bing. I'm not sure what you mean by "is order of growth and time complexity same". Do you want to know if these two terms are interchangeable? Or do you want to know how they are related? Or do you want to know how to calculate them for a given algorithm?

According to the web sources I found, time complexity is a measure of how long it takes an algorithm to run as a function of the input size . It is usually expressed using the Big O notation, which describes the worst-case scenario of an algorithm's performance. For example, O(n) means that the algorithm's running time is linearly proportional to the input size n.

Order of growth is a way of describing the approximate behavior of the time complexity function of an algorithm when the input size is very large . It is also expressed using the Big O notation, but it ignores constant factors and lower-order terms that do not affect the overall trend of the function. For example, if an algorithm has a time complexity of 3n^2 + 5n + 2, its order of growth is O(n^2), because n^2 is the dominant term when n is very large.

So, order of growth and time complexity are not exactly the same, but they are closely related. Order of growth is a simplified way of expressing time complexity that focuses on the most significant factor that affects the algorithm's performance. I hope this helps. 😊

is order of growth and time complexity same ?