Twitter Thread by Yosra ■ looking for SWE internship

Yosra ■ looking for SWE internship
@yosracodes

■ Space and Time Complexity ■

One of the most important concepts when studying DSA is the complexity of the algorithm

In this thread I will dive deep into what that means and how you can measure it ■■

When learning algorithms you can measure how efficient the algorithm is by knowing how many lines of code it executes to run an algorithm for an input of size n.

However, it is very hard to measure exactly how many lines of code the algorithm takes.

This is why big O exists.

what is big O? ■

Big O is the most used asymptotic notation. Asymptotic notations are a way to approximate how many lines of code the algorithm executes relative to the input size which we will refer to as "n" for the rest of the thread

To understand Big O better I will explain it through an algorithm: Selection Sort

This algorithm sorts an array by looping through the whole array and for each element in the array it iterates through the rest of the array to find the smallest element to put it in the beginning

This image visualizes how selection sort works:

in this algorithm we consider the list as the input and n is the size of that list aka the number of elements inside it.

Assume that the if condition takes constant time to be executed.

To know the time complexity we must know how many times the if statement is executed

let's study the inner loop

first when i = 1, the inner loop is executed n times, when i is incremented the inner loop is exectured n-1 times... and so on

if we add them together we will find that the inner loop is executed:

n + n-1 + ... + 2 + 1 times

This is a geometric series!

if we calculate it we will get: n(n-1)/2 which is equal to $n^2/2 - n/2$

when we calculate big O notations we only care about the dominant terms and we do not care about the coefficients

in that case we will remove the coefficients so we will get: $n^2 - n$ and since $n^2 > n$ (n^2 is more dominant), we take n^2 as our big O notation

therefore, the time complexity of Selection Sort is $O(n^2)$!

Here is a video that explains time complexity in a more detailed way made by @hackerrank
https://t.co/cH0CQjJCqX

■ How about space complexity?

Space Complexity is the amount of memory used by the algorithm

if the algorithm needs constant extra space (i.e the space taken is not affected by n) to perform the algorithm its space complexity is O(1)

in our selection sort example we did the sort in place (i.e the original array is changed to a sorted one) so the space complexity is O(1)

what if we want to keep the original array and create a new one to store the sorted values?
In this case the space complexity will be O(n).

■ Big O is not the only asymptotic notation out there but it is the most used one.

Want to check out more asymptotic notations? Check out this article from @geeksforgeeks
https://t.co/JxfHb7q5Cw

That's it! if you want to ask any questions my dms are open!

Did you like this thread?
I would appreciate it if you:
- retweet the first tweet

- follow me

I post threads like these and other programming related content :D