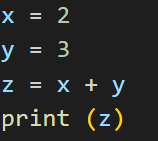
Complexity Classes in Big O

Increasing Order Of Complexity

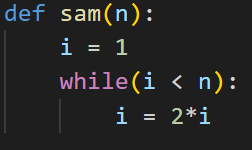
1. Constant Time Complexity (Best One ) O(1) :

This is the best time complexity , and everybody want this Complexity , this will achieve only in our code there is no loop Only Statements



1. Logarithmic Time Complexity O(log n) :

Here this one best compare to other and worst compare to Constant O(1) , It will decrease based on the log base



Now we see n = 64 , start with I = 1 it will go and increment I = 2 then I = 4 , I = 8 , I = 16 , I = 32 , if I = 64 our condition is false

How many time our code runs , 6 time

If N is :

64 🡪 6 time

32 🡪 5 time

16 🡪 4 time

If we see log of 64 when base is 2 == 6;

Log64(base2) == 6;

Log32(base) == 5 ;

We will Use Binary Search Using Log n

1. Linear Time Complexity O(n):

We will Using This is Array Searching Algorithm

Ex : Linear Search

4 . Quadratic Time Complexity O(n2) :

We will use Sorting Algorithm Specifically Merge Sort

1. Cubic Time Complexity O(n3) :

In Matrix Multiplication we use This One

Whenever we apply Three Internal For Loop We use this Time Complexity

1. Polynomial Time Complexity O(nc) :

It means Order(N Power C)

C means Constant value which is greater than 0.\

C > 0

1. ExPonential Time Complexity O(Cn):

It Means Order Of (C to the Power N)

C ithe Constant value is greater than 1

C > 1

The Relationship between 2n (2 Power N)

, n! (n factorial) , nn(n Power n)

2n less than N!(2N < Nn)

N! less than nn (n! < nn)

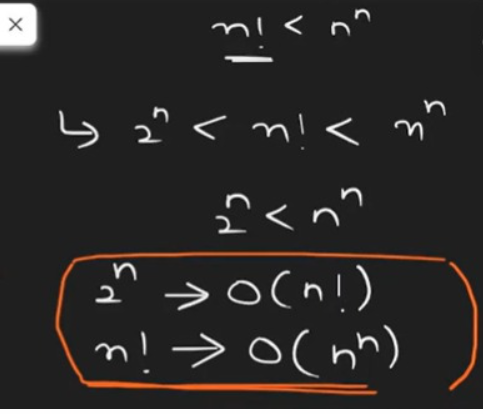
2n < N! < Nn

In Big O Notation we use Worst case(Larger Value)

So we can write

2n 🡪 O(n!)

N! 🡪 O(Nn)



So has wee Saw Many Time Complexity For all these

Constant O(1) is Best and Exponential O(Cn) is Worst

If Two People Writing same Code for given Problem first one gives Cubic Time Complexity O(n3) Second Givews O(n) , So We Take the Second person code

O(n3) > O(n)

Someone gives 2 power n (2n) and 3 Power n (3n) which one is Greater

3(n) > 2(n)

Likewise (logn)2 Is Lesser than N

But (Log n)log n > Greater than N

If we have log base2 n and log base3 n which one is Greater

Always which base is low than is Greater one

Log 2 n > log 3 n