Omega Notation (Ω)

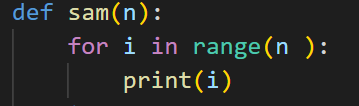
Whenever We Develop An Algorithm Not only on Data Structure like ML Algorithm Or Any Domain we Use Big O (O) To Calculate whether the Algorithm is Effective or Not , It Is Work Good On Worst Case Scenario Also(Huge Data) . So Big O Notation we used as Upper Bond Large Data

Why we use Omega Notation :

Omega Notation Helps to Calculate the Lower Bond (Low Data ).

It will be on Best Case Scenario . So that’s Why we don’t use this for Mostcase

Ex :



If we See the Code Based on the n it takes time .

If n is 10 the value is quite low so we called it as a Lower Bond .

If n is 10000000000 the value is huge we called as a upper bond .

We all know for upper bond it takes Some More time

Math For Omega :

If we See on Big O

F(x) <= O(constant)g(x)

So in Big o F(x) is less than or equal to Constant of g(x).

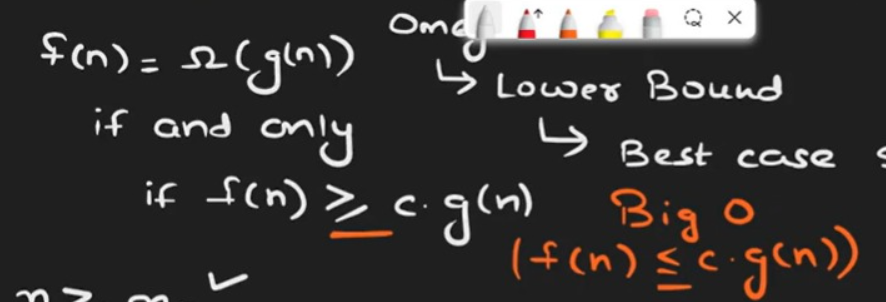
For Omega Is Different

F(n) >= Ω (constant) g(n)

So in Omega F(n) Greater than or Equal to g(n)

In Big O Our Constant is Greater than 0

But in Omega Constant Greater than 0



See the Example :

1. f(n) = n

g(n) = 5n

f(n) = Ω g(n)

f(n) > = c \* g(n)

n > = c \* 5n

What c value we Assign if this Statement Gets True

Note : C is Greater than 0

We use c = ½ (half)

Now our Condition is Satisfied

1. f(n) = 5n ; g(n) = n

f(n) =cg(n)

f(n) > = c \* g(n)

5n >= c\* g(n)

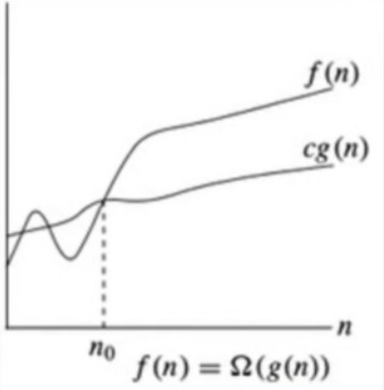
So here we put c value has 5 or less than 5 greater than 0 it is Satisfied .

1. F(n) = n2 ; g(n) = n2 + n + 10

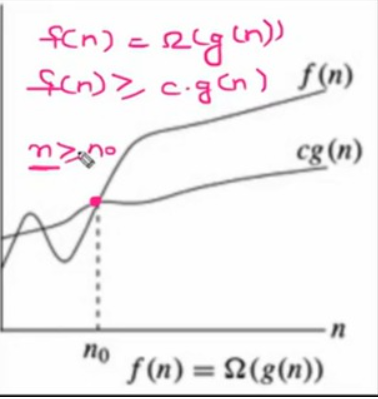
So here we see if n2 > c \* (n2 + n + 10 )

Our condition is satisfied .

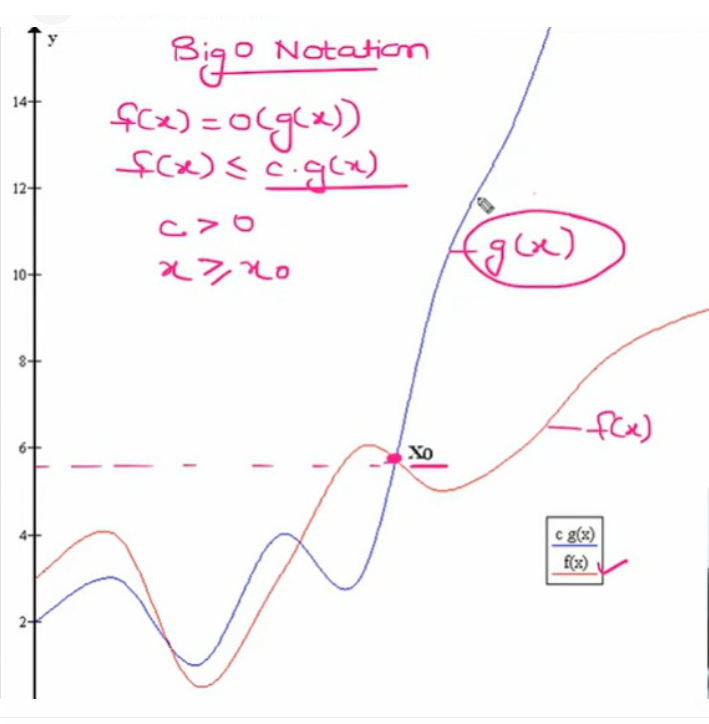
Graph Structure For Omega Notation :



If we see Big O Notation at one point X(n) is greater x0(n0) like Same in omega



This is Omega



This is Big O