

# **Big $O$ Notation**

**Hiten Dalmia**

B. Sc. (Hons.) Mathematics, 1st year

2232174

## Contents

**Abstract.** In this paper we introduce the Big  $O$  notation, explain the concept and its uses, particularly in studying growth of sequences and in computer science. We also discuss some examples where the concept is used along with its advantages.

## 1 Introduction

Big  $O$  notation is a mathematics symbolism or notation that is used to describe the limiting behaviour of a function when the argument of the function tends towards a particular value like infinity (asymptotic behaviour). In simpler terms, it is a way of denoting the rate of a function's growth or decline.

Big  $O$  notation is widely used in mathematics and computer science. In computer science, big  $O$  notation is used to measure the run time required by algorithms as the size of their input increases.

## 2 A History



Figure 1: Paul Gustav Heinrich Bachmann (1837-1920)

Big  $O$  is a member of a family of notations invented by Paul Bachmann (1837-1920), Edmund Landau (1877-1938), and others, collectively called Bachmann–Landau notation or asymptotic

notation. The letter  $O$  was chosen by Bachmann to stand for *Ordnung* (a German word), meaning the order of approximation.



Figure 2: Edmund Georg Hermann Landau (1877-1938)

The symbol  $O$  was first introduced by number theorist Paul Bachmann in 1894, in the second volume of his book *Analytische Zahlentheorie* ("analytic number theory"). The number theorist Edmund Landau adopted it, and was thus inspired to introduce in 1909 the notation  $o$ ; hence both are now called Landau symbols. These notations were used in applied mathematics during the 1950s for asymptotic analysis.

As for the other Landau symbols, The symbol  $\Omega$  (in the sense "is not an  $o$  of") was introduced in 1914 by Hardy and Littlewood. Hardy and Littlewood also introduced in 1916 the symbols  $\Omega_R$  ("right") and  $\Omega_L$  ("left"), precursors of the modern symbols  $\Omega_+$  ("is not smaller than a small  $o$  of") and  $\Omega_-$  ("is not larger than a small  $o$  of").

In the 1970s the big  $O$  was popularized in computer science by Donald Knuth, who introduced the related Theta notation, and proposed a different definition for the Omega notation.

### 3 Definition

Let  $f$ , the function to be estimated, be a real or complex valued function and let  $g$ , the comparison function, be a real valued function. Let both functions be defined on some unbounded subset of the positive real numbers, and  $g(x)$  be strictly positive for all large enough values of  $x$ . One writes

and it is read " $f(x)$  is big  $O$  of  $g(x)$ " if the absolute value of  $f(x)$  is at most a positive constant multiple of  $g(x)$  for all sufficiently large values of  $x$ . That is,  $f(x) = O(g(x))$  if there exists a positive real number  $M$  and a real number  $x_0$  such that

$$|f(x)| \leq Mg(x) \quad \text{for all } x \geq x_0.$$

In many contexts, the assumption that we are interested in the growth rate as the variable  $x$  goes to infinity is left unstated, and one writes more simply that

$$f(x) = O(g(x)).$$

The notation can also be used to describe the behavior of  $f$  near some real number  $a$  (often,  $a = 0$ ): we say

$$f(x) = O(g(x)) \quad \text{as } x \rightarrow a$$

if there exist positive numbers  $\delta$  and  $M$  such that for all defined  $x$  with  $0 < |x - a| < \delta$ ,

$$|f(x)| \leq Mg(x).$$

As  $g(x)$  is chosen to be strictly positive for such values of  $x$ , both of these definitions can be unified using the limit superior:

$$f(x) = O(g(x)) \quad \text{as } x \rightarrow a$$

if

$$\limsup_{x \rightarrow a} \frac{|f(x)|}{g(x)} < \infty.$$

And in both of these definitions the limit point  $a$  (whether  $\infty$  or not) is a cluster point of the domains of  $f$  and  $g$ , i. e., in every neighbourhood of  $a$  there have to be infinitely many points in common. Moreover, the  $\limsup_{x \rightarrow a}$  (at least on the extended real number line) always exists.

In computer science, a slightly more restrictive definition is common:  $f$  and  $g$  are both required to be functions from some unbounded subset of the positive integers to the non-negative real numbers; then  $f(x) = O(g(x))$  if there exist positive integer numbers  $M$  and  $n_0$  such that

$$f(n) \leq Mg(n) \quad \text{for all } n \geq n_0.$$

### 3.1 Graphical Representation

In this figure the function to be estimated is  $f(x)$ , the red graph. The comparison function we are using is  $g(x)$ . We have a positive real number  $M$  such that for all values of  $x \geq x_o$ , we get

$$|f(x)| \leq Mg(x)$$

We can see in figure ?? that for values of  $x \geq x_o$  the shape of the function  $Mg(x)$  imitates that of  $f(x)$ . It is easy to observe graphically through the given figure how as  $x \rightarrow \infty$ , the functions  $f(x)$  and  $O(g(x))$  will approach each other, giving us

$$f(x) = O(g(x)) \quad \text{as } x \rightarrow \infty.$$

## 4 Usage

### 4.1 In Mathematics

In most cases, big  $O$  notation is used for asymptotic cases, that is, as  $x \rightarrow \infty$ . In such cases, the terms that grow faster than the others overshadow the effect of the other terms. Hence we formulate a few simple rules to dealing with functions in asymptotic cases.

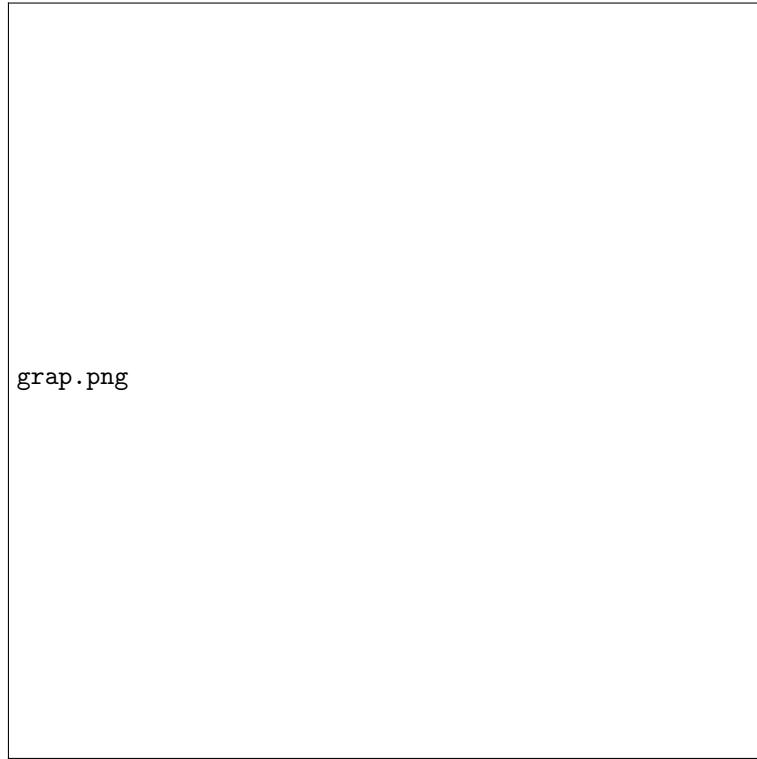


Figure 3: A graphical representation

1. If  $f(x)$  is a sum of several terms, if there is one with largest growth rate, it can be kept, and all others omitted.
2. If  $f(x)$  is a product of several factors, any constants (factors in the product that do not depend on  $x$ ) can be omitted.

Let us take a function

$$f(x) = 6x^4 + 2x^3 + 5$$

now to describe its growth rate using big  $O$  notation, we first simplify it as such. The function has three terms  $6x^4$ ,  $2x^3$  and 5. Out of these three as  $x \rightarrow \infty$  it is obvious that  $6x^4$  will be the fastest growing term as it has the highest exponent power. Here we use the first rule.

Now, in the term  $6x^4$ , we disregard the 6 as per the second rule that we have discussed above. We now have the simplified form,  $x^4$ . Thus we say, " $f(x)$  is a big  $O$  of  $x^4$ ". Mathematically we write,

$$f(x) = O(x^4)$$

Applying the formal definition of the big  $O$  notation, the statement

$$f(x) = O(x^4)$$

is equivalent to

$$|f(x)| \leq Mx^4$$

from some  $x_o \in \mathbb{R}$ , a positive real number  $M$  and  $\forall x_o > x$ .

To show this let us take  $x_0 = 1$  and  $M = 13$ .

Using triangle inequality, for all  $x_0 > x$ ,

$$\begin{aligned} |6x^4 - 2x^3 + 5| &\leq 6x^4 + |2x^3| + 5 \\ &\leq 6x^4 + 2x^4 + 5x^4 \\ &= 13x^4. \end{aligned}$$

Hence

$$|6x^4 - 2x^3 + 5| \leq 13x^4.$$

#### 4.1.1 Error Terms or Infinitesimal Asymptotics

In mathematics, big  $O$  notation is often used to describe how closely a finite series approximates a given function, especially in the case of a truncated Taylor series or asymptotic expansion. Given below is an example of its use in the Taylor series of  $e^x$ .

Big  $O$  can also be used to describe the 'error term' in an approximation to a mathematical function. The most significant terms are written out explicitly, and then the least-significant terms can be summarized in a single big  $O$  term. In the expansion of  $e^x$ , two expressions of it that are valid when  $x$  is small are:

$$\begin{aligned} e^x &= 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots \quad \text{for all } x \\ &= 1 + x + \frac{x^2}{2} + O(x^3) \quad \text{as } x \rightarrow 0 \\ &= 1 + x + O(x^2) \quad \text{as } x \rightarrow 0 \end{aligned}$$

What these expressions mean is that the when  $x \rightarrow 0$ , the value of the error by ignoring the terms of lesser significance is equal to their big  $O$  notation.

For example in the third expression, we are saying

$$\begin{aligned} e^x &= 1 + x + O(x^2) \quad \text{as } x \rightarrow 0 \\ |e^x - (1 + x)| &\leq Mx^2 \quad \text{as } x \rightarrow 0 \end{aligned}$$

for some positive  $M \in \mathbb{R}$ .

## 4.2 In Computer Science

In computer science, big  $O$  notation is useful for analyzing the 'complexity' of algorithms. Let us say we have a problem of size  $n$  and the time taken to complete the problem is given by  $T(n) = n^2 - n - 1$ . As discussed above as  $n \rightarrow \infty$  the terms other than  $n^2$  will become insignificant.

One good way to understand this is even if  $n$  had a very large coefficient, say, 20000, i.e.,  $20000n$ . As the value of  $n$  exceeds 20000, only  $n^2$  will be significant as its value will be much greater than the other terms.

So similar to what is done in mathematics, we denote the complexity of the algorithm as

$$T(n) \in O(n^2).$$

Unlike the case before we do not use the '=' sign here, this is because here  $T(n)$  is the *exact* complexity of that particular algorithm and  $n^2$  is only meant to be an upper bound of the complexity, ideally the least upper bound.

We also call the complexity of an algorithm its 'order'. Listed below are some common orders for algorithms

1. A constant-time algorithm (a sequence of statement) is of order 1, i.e.,  $O(1)$ .
2. A linear-time algorithm (a loop) is of order  $n$ , i.e.,  $O(n)$ .
3. A linear-time algorithm (a singly nested loop) is of order  $n$ , i.e.,  $O(n)$ .
4. There also exist complexities like  $O(\log(n))$ , an example of this would be the binary search algorithm.

## 5 Other Landau Notations

### 5.1 Little-o notation

Similar to the big  $O$  notation,  $f(x)$  is  $o(g(x))$  means that  $g(x)$  grows faster than  $f(x)$ . Here,  $f(x)$  is  $o(g(x))$ , is read as, "f(x) is little-o of g(x)".

Let  $f$  be a real or complex valued function and  $g$  a real valued function, both defined on some unbounded subset of the positive real numbers, such that  $g(x)$  is strictly positive for all large enough values of  $x$ . One writes

$$f(x) = o(g(x)) \quad \text{as } x \rightarrow \infty$$

if for any  $\epsilon > 0$  there exists a constant  $x_0$  such that

$$|f(x)| \leq \epsilon g(x) \quad \text{for all } x \geq x_0.$$

A few examples:

$$7x^2 = o(x^3) \quad \text{as } x \rightarrow \infty$$

$$\frac{1}{x} = o(1) \quad \text{as } x \rightarrow \infty$$

The key difference between  $O(g(x))$  and  $o(g(x))$  is that in the case of big  $O$  we need the defined condition to be true for at least one positive real  $M$ , whereas, in the case of little-o we need the defining condition to hold true for every positive real number  $\epsilon$ . For instance,

$$7x^2 = O(x^2) \quad \text{as } x \rightarrow \infty$$

however

$$7x^2 \neq o(x^2) \quad \text{as } x \rightarrow \infty.$$

## 5.2 Big Omega notation

Another member of the Landau family of notations is the big omega,  $\Omega$ . There are two commonly used definitions of this notations that mutually exclusive to each other. We will look at both of them.

### 5.2.1 The Hardy-Littlewood Definition

This definition is mostly used in mathematics particularly in analytic number theory. It is defined as follows,

$$f(x) = \Omega(g(x)) \text{ as } x \rightarrow \infty \text{ if } \limsup_{x \rightarrow \infty} \left| \frac{f(x)}{g(x)} \right| > 0.$$

We could also consider  $f(x) = \Omega(g(x))$  to be the negation of  $f(x) = o(g(x))$ .

Later on, two new symbols were introduced,  $\Omega_R$  and  $\Omega_L$ , defined as:

$$f(x) = \Omega_R(g(x)) \text{ as } x \rightarrow \infty \text{ if } \limsup_{x \rightarrow \infty} \left| \frac{f(x)}{g(x)} \right| > 0.$$

$$f(x) = \Omega_L(g(x)) \text{ as } x \rightarrow \infty \text{ if } \limsup_{x \rightarrow \infty} \left| \frac{f(x)}{g(x)} \right| < 0.$$

For example,

$$\begin{aligned} \sin x &= \Omega(1) && \text{as } x \rightarrow \infty \\ \sin(x) + 1 &= \Omega_L(1) && \text{as } x \rightarrow \infty \\ \sin x &= \Omega_R(1) && \text{as } x \rightarrow \infty \end{aligned}$$

### 5.2.2 The Knuth Definition

In 1976 Donald Knuth published a paper to justify his use of the  $\Omega$ -symbol to describe a stronger property. Knuth wrote: "For all the applications I have seen so far in computer science, a stronger requirement ... is much more appropriate". He defined

$$f(x) = \Omega(g(x)) \Leftrightarrow g(x) = O(f(x))$$

We can see that this is mutually exclusive to the Hardy-Littlewood definition as

$$g(x) = O(f(x)) \Rightarrow f(x) \neq o(g(x))$$

## 6 Teaching Calculus with Big $O$

In the June/July 1998 edition of the 'Notices of the AMS', Donald E. Knuth, talks about an alternative pedagogy for calculus. This pedagogy, which he calls 'O Calculus', involves using the big  $O$  notation to teach the ideas of calculus. It begins as such.

We start the course by defining a simpler ' $A$  notation' which means 'absolutely at most'. For example,

$$f(x) = A(2)$$

implies that the absolute value of  $f(x)$  is less than or equal to 2. For one, this notation naturally goes well with decimals. As an example,

$$\pi \approx 3.14$$

is the same as saying

$$\pi = 3.14 + A(0.002).$$

To get familiar with this new notation we will evaluate a few expressions involving the  $A$  notation.

$$10^{A(2)} = A(100)$$

$$\begin{aligned} (3.14 + A(.005))(1 + A(0.01)) &= 3.14 + A(.005) + A(0.0314) + A(.00005) \\ &= 3.14 + A(0.03645) \\ &= 3.14 + A(.04). \end{aligned}$$

One must realise that the equality operation here is not symmetric.

$$3 = A(5) \text{ and } 4 = A(5) \not\Rightarrow 3 = 4.$$

We now define the big  $O$  notation in terms of the  $A$  notation. In its simplest form,  $O(x)$  stands for something that is  $CA(x)$  for some positive constant  $C$ .

Before defining the derivative, we would first define a 'strong derivative': A function  $f$  has a strong derivative  $f'(a)$  at the point  $a$  if it satisfies the following condition:

$$f(a + \epsilon) = f(a) + f'(a)\epsilon + O(\epsilon^2).$$

And the derivative can be defined with the used of the "o notation", which is the equivalent of taking limits, to give the general definition

$$f(a + \epsilon) = f(a) + f'(a)\epsilon + o(\epsilon).$$

## 7 Bibliography

1. N. G. de Bruijn (1958) - Asymptotic Methods in Analysis, Amsterdam: North-Holland.
2. Donald Knuth (June–July 1998) - "Teach Calculus with Big O", Notices of the American Mathematical Society.
3. MIT - 16.070 Big *O* Notation,  
[https://web.mit.edu/16.070/www/lecture/big\\_o.pdf](https://web.mit.edu/16.070/www/lecture/big_o.pdf)
4. Growth of sequences — OEIS (Online Encyclopedia of Integer Sequences) Wiki,  
[https://oeis.org/wiki/Growth\\_of\\_sequences](https://oeis.org/wiki/Growth_of_sequences)
5. UCSC - CSE 102: Introduction to Analysis of Algorithms,  
<https://classes.soe.ucsc.edu/cse102/Fall21/Handouts/AsymptoticGrowth.pdf>
6. Khan Academy - Big-O notation,  
<https://www.khanacademy.org/computing/computer-science/algorithms/asymptotic-notation/a/big-o-notation>