

Function growth and Big O notation

Even though functions aren't limited to mappings from numbers to numbers, this is certainly one of the most commonly encountered types of functions. We can classify these functions based on how quickly the output grows relative to the input.

Constant functions

In a constant function, the output is always the same independent of the argument. They follow the pattern $f(x) = c$, where c is some fixed value.

Example All of the following are constant functions.

1

- $f(x) = 0$
- $f(x) = 17,033,581,395$
- $f(x) = e^\pi$
- $f(x) = 10^{10^{10}}$

Linear functions

Linear functions follow the general pattern $f(x) = kx + d$. They can be visualized as a line in a Cartesian coordinate system.

Example All of the following are instances of linear functions.

2

- $f(x) = 5x + 10$
- $f(x) = x - \sqrt{2}$
- $f(x) = \frac{9x}{4}$

Exercise Explain why every constant function is also a linear function.

1

Polynomial function

In a polynomial function, the argument is multiplied with itself a fixed number of times. This is expressed as exponentiation.

Example All of the following functions are polynomials.

3

- $f(x) = x^2$
- $f(x) = x^3 - 5x$
- $f(x) = -5x^4 + 7x^2 + (2x + 3)^2$
- $f(x) = \frac{3x^4 + x^3}{17}$

When plotted in a coordinate system, polynomials aren't straight lines but curves. Polynomial functions grow faster than linear functions, but a linear function may initially return higher values. Only as we move towards larger and larger numbers is a polynomial guaranteed to eventually outstrip a linear function.

Example 4 Consider the linear function $f(x) = 1000x$ and the polynomial function $g(x) = x^2 - 9x$. The table below shows the growth of those two functions.

x	$f(x)$	$g(x)$
0	0	0
1	1000	-8
2	2000	-14
3	3000	-18
4	4000	-20
5	5000	-20
6	6000	-18
7	7000	-14
8	8000	-8
9	9000	0
10	10,000	10
100	100,000	9,100
1,000	1,000,000	991,000
5,000	5,000,000	24,955,000

Formally, a function is polynomial iff it can be rewritten to fit the template below, where k_0, \dots, k_n are arbitrary numbers.

$$f(x) = k_n x^n + k_{n-1} x^{n-1} + \dots + k_1 x^1 + k_0 x^0,$$

Exercise 2 Rewrite the formulas from the previous example so that they fit this pattern. For instance, $f(x) = x^2$ can be rewritten as $f(x) = 1x^2 + 0x^1 + 0x^0$.

Reminder: $(a + b)^2 = a^2 + 2ab + b^2$

Exercise 3 Explain why every linear function is also a polynomial function.

Exponential functions

Like polynomial functions, exponential functions involve exponentiation. But in a polynomial the argument of the function acts as the base, whereas in an exponential function it serves as the exponent.

Example 5 All of the following are exponential functions.

- $f(x) = 2^x$
- $f(x) = 5 \times (10^x)$
- $f(x) = 3.7 \times (1^x)$

Exponential functions grow even faster than polynomials, but once again this may only hold as we consider larger and larger values for x . A polynomial function may initially return larger values than an exponential function.

Exercise 4 Give an example of a polynomial function f and an exponential function g such that $f(x) > g(x)$ for relatively small values of x (e.g. $x < 100$)

but $f(x) < g(x)$ for larger values.

Big O notation

As one considers larger and larger values for x , some parts of a function become increasingly irrelevant.

Example 6 Consider the function $f(x) = x^3 + x^2$. For small x , x^2 contributes a significant proportion of the output value, 10% or more. But as we increase x , the contribution of x^2 becomes negligible, until it doesn't even make up 1%.

x	x^3	$x^3 + x^2$
0	0	0
1	1	2
2	8	12
3	27	36
4	64	80
5	125	150
6	216	252
7	343	392
8	512	576
9	729	810
10	1000	1100
100	1,000,000	1,010,000
1,000	1,000,000,000	1,001,000,000
5,000	125,000,000,000	125,025,000,000

Often we do not really care about the precise value returned by a function, we just want to know the rough ballpark and how quickly the function grows. The Big-O notation was designed just for this purpose. Instead of a complicated function like $f(x) = 15x^5 + \frac{3x^4}{6} - 500,000x$, we can just write $O(x^5)$ to signal that even though the full function may be more complicated, its overall growth pattern will resemble $f(x) = x^5$ as we start looking at larger and larger values for x .

While the Big-O notation has a fairly involved definition, the intuition is really simple: find the fastest-growing component in the function's definition and remove everything else. The general principle is that exponential trumps polynomial trumps linear trumps constant.

$f(x)$	Big-O
$3x + 100$	$O(3x)$
$50x - x^2$	$O(x^2)$
10	$O(10)$
$2^x - x^{100}$	$O(2^x)$

The Big-O examples in the right column might strike you as very incomplete representations of f , but that's the point: we use the Big-O notation when we do not

care about the minor details of f and just want to see the overall growth pattern. In fact, things are often simplified even further: constants are replaced with just 1, and in linear functions we drop the multiplier. The table above then becomes the one below.

$f(x)$	Big-O
$3x + 100$	$O(x)$
$50x - x^2$	$O(x^2)$
10	$O(1)$
$2^x - x^{100}$	$O(2^x)$

We can now relate each type of function to a specific type of Big-O expression.

- **constant:** $O(1)$
- **linear:** $O(x)$
- **polynomial:** $O(x^2), O(x^3), \dots$
- **exponential:** $O(2^x), O(3^x), \dots$

Exercise 5 For each one of the formulas below, right down its simplified version in Big-O notation.

1. $f(x) = x \times x$
2. $f(x) = x(3x + 5)$
3. $f(x) = x^2 + 17(x - \frac{2^x}{x})$
4. $f(x) = 2^x + 17 - (1 + 2^x)$

Recap

- For numerical functions, we can distinguish four subtypes: constant, linear, polynomial, exponential
- constant: $f(x) = c$
- linear: $f(x) = kx + d$
- polynomial: $f(x) = k_n x^n + k_{n-1} x^{n-1} + \dots + k_1 x^1 + k_0 x^0$
- exponential: $f(x) = k(b^x)$
- The Big-O notation shows the part of the function $f(x)$ that contributes most to the output as we consider larger and larger values for x .