

# Complexity

---

Different implementations of the same problem may be faster or slower than one another. We can test this by analysing the runtime for random input, but that requires us to implement each approach, and create a testing harness. Also, performance may depend on the implementation in a particular language, and comparison requires us to test in the same hardware and software environment.

## Basic Steps

We can consider the following to be basic steps, and count them as a measure of how much work our program is doing:

- reading the value of a variable
- assigning a value to a variable
- simple arithmetic operations
- comparing the values of two variables
- calling a function
- returning a value
- reading the value at a given index in a Python list

We expect each step to take approximately a constant time.

The number of steps for an algorithm may grow as the size of the input grows.

Big-O notation gives us a way of quantifying this and giving an upper bound.

## Big-O Notation

$f(x)$  is  $O(g(x))$  if:

$$(x > k) \rightarrow (f(x) \leq Cg(x))$$

- there are two constants  $k$  and  $C$

- such that when  $x > k$ ,  $f(x) \leq Cg(x)$