For a few programs, the cost is fixed and can be calculated by examining the program text. Moreover, frequently, however, cost depends on characteristics of the input, such as length.

• \(n^2\) steps, rather than \(3n\) versus \(2n\) steps.
• \(n^2\) versus \(n^3\).
• When comparing programs (or algorithms), you should first pay attention to gross differences in time or space consumed, for example, instead of seconds and bits (bits). Frequently, we abstract our units, and measure steps and objects.

A running program consumes resources such as time (seconds) and space.

Accounting for the resources used by computations
term.

where \( k \) is some constant. They don’t grow faster than \( k n^2 \), i.e., \( u^2 \) equals \( u \) plus some terms that are of order \( u \).

\[
(1) \quad (u^2)O + u = d
\]

For example, if we say

\[
(u)O
\]

is less than (or some multiple of the function \((u)f\). For example, \( (u)O \) is simply means that the cost function is bounded by

The \text{Big-O} notation simply means that the cost function is bounded by

\text{Big-O} denotes

\text{order-of-magnitude} of the cost. The notation used is sometimes called

\text{arithmetic}
(4) \((u)f \omega \preceq (u)\eta \preceq 0\)

exist positive integers \(\omega\) and \(\eta\) such that

In other words, \(((u)f)O\) is the set of all functions \((u)\eta\) such that there

for all \(\eta \preceq u\).

(3) \((u)f \omega \preceq (u)b \preceq 0\)

if there are positive integers \(\omega\) and \(b\) such that

(2) \(((u)f)O = (u)b\)

More precisely, \(((u)f)O\) is said to be \((u)b\) written

Definition. A function, \((u)f\), is said to be \((u)b\) written

Order Arithmetic (cont.)
Orden Arithmetic (cont.)
(7) \[(\mathcal{O}u)O = u + \cdots + 3 + 2 + 1\]

(6) \[(u)O + \frac{\mathcal{O}}{\mathcal{O}u} = u + \cdots + 3 + 2 + 1\]

(5) \[\frac{\mathcal{O}}{u} + \frac{\mathcal{O}}{\mathcal{O}u} = \frac{\mathcal{O}}{(1 + u)u} = u + \cdots + 3 + 2 + 1\]

For example,

Order Arithmetic (cont.)
(II) \[ ((u)\delta \times (u)f)O = ((u)\delta)O \times ((u)f)O \]

(II) \[ ((u)f)O = ((u)f)O + ((u)f)O \]

(6) \[ ((u)f)O = ((u)f)O \times X \]

(8) \[ ((u)f)O = (u)f \]

• Here are some equivalences that allow you to manipulate equations:

Order Arithmetic (cont.)
changes the value by a constant factor of \( \log_2 c \).

Also, the base to which a logarithm is computed doesn’t affect the order

Order Arithmetic (cont.)