Lesson 1: Solving Problems with Algorithms

Creation Emerging from the Collapse of Wholeness to a Point

Wholeness of the Lesson

This course on Algorithms provides tools for designing and implementing highly efficient algorithms to solve problems. Tools include methods for determining the level of efficiency of an algorithm, for determining which algorithm among many that solve the same problem is optimal, and for demonstrating that an algorithm is correct. To develop these tools we begin working with the tools we already have and systematically refine and enhance them. Transformation of consciousness in any domain of life must begin with where one is and achieves the goal through many rounds of refinement.

What Is an Algorithm?

- An algorithm is a step-by-step procedure for solving a problem
- Algorithms can be implemented in programming languages like Java and C
- As programmers, we already are familiar with creation of algorithms, testing their correctness, and measuring their efficiency
- A course in algorithms refines these understandings
- The starting point is to try solving several problems with the skills we currently have and discover ways to improve our ability to
  - Determine efficiency of an algorithm
  - Determine correctness of an algorithm
  - Figure out ways to improve an algorithm and verify when we have genuinely made an improvement
  - Communicate the concept of an algorithm without relying on code

Today's Lesson

- Review the mathematical preliminaries for the course
- Go over problem list that you will be working on for the next two days.

Important Math Review Points

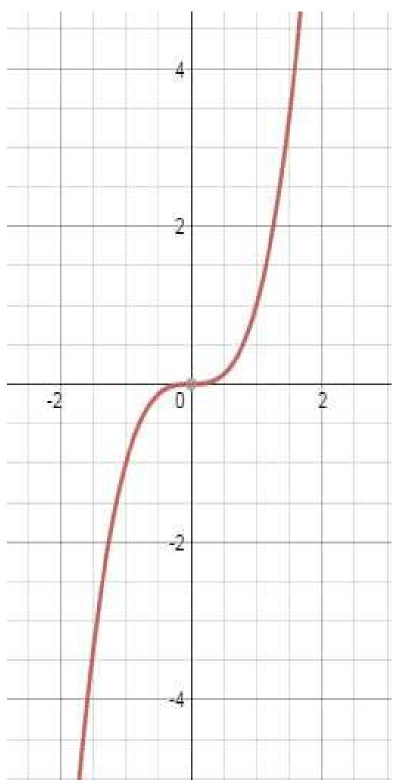
April 15, 2020

1 Increasing/Nondecreasing functions

A function is increasing if its graph climbs steadily upward. More precisely:

Defintion. A function f on the real line is *increasing* (*nondecreasing*) if, whenever $x_1 < x_2$ ($x_1 \leq x_2$), $f(x_1) < f(x_2)$ ($f(x_1) \leq f(x_2)$).

Example: $f(x) = x^3$ is an increasing function.



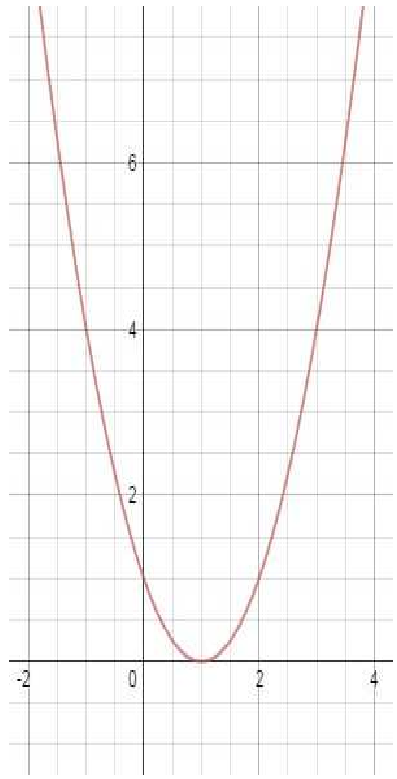
Question Is $f(x) = x^2$ increasing?

2 Eventually Nondecreasing Functions

A function is eventually nondecreasing if for all values beyond a certain point on the x-axis, the graph steadily climbs. More precisely,

Definition. A function f is *eventually nondecreasing* if for some real number x_0, f is increasing on $[x_0, \infty)$. In other words, for some x_0 we have that whenever $x_0 \leq x_1 \leq x_2$, then $f(x_0) \leq f(x_1) \leq f(x_2)$.

Example: $f(x) = (x-1)^2$.

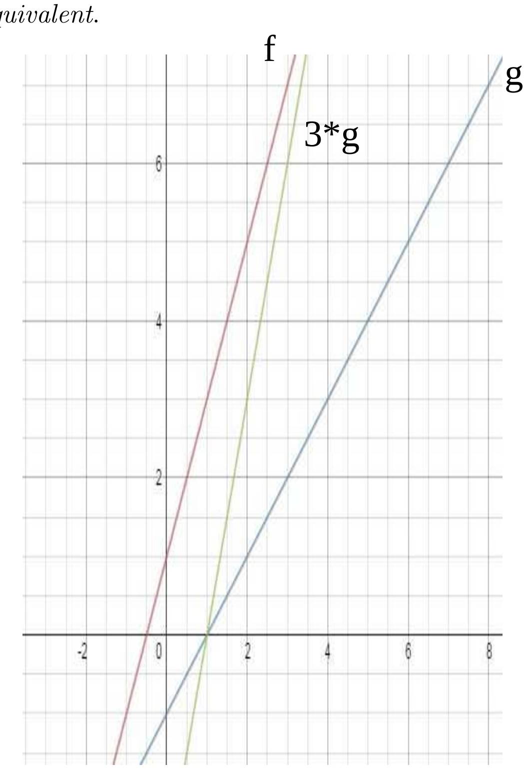


3 Growth Rates of Functions

Sometimes a function grows faster than another. Sometimes a function only *appears* to grow faster than another.

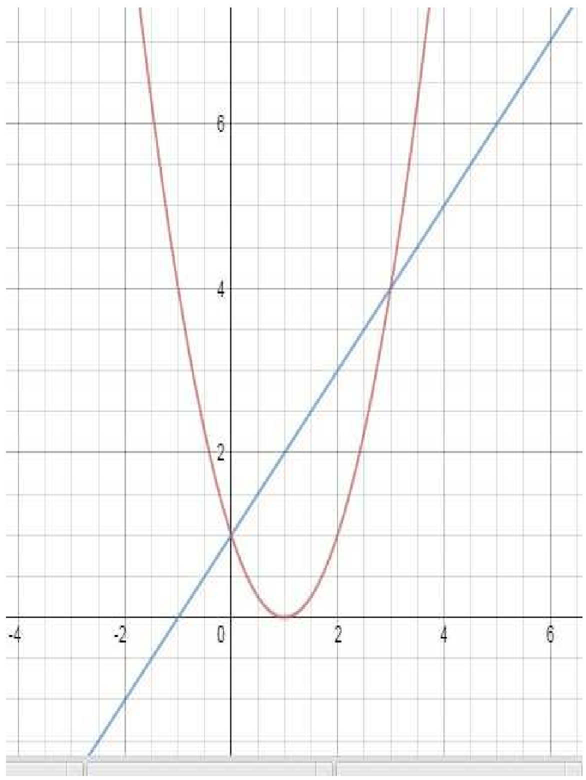
Example (Same Growth Rate) $f(x) = 2x + 1$ and $g(x) = x - 1$. In the graph, f appears to grow faster since the graph of f eventually lies above the graph of g . But if we multiply g by 3, we get a new function $3x - 3$, and this one's graph eventually lies above that of f.

Basic Idea. If we have functions f and g and if we can find constants c and d so that $cf(x)$ eventually lies above $g(x)$ and $d(g(x))$ eventually lies above $f(x)$, we say that f and g *have the same growth rate* and also that f and g are *asymptotically equivalent.*

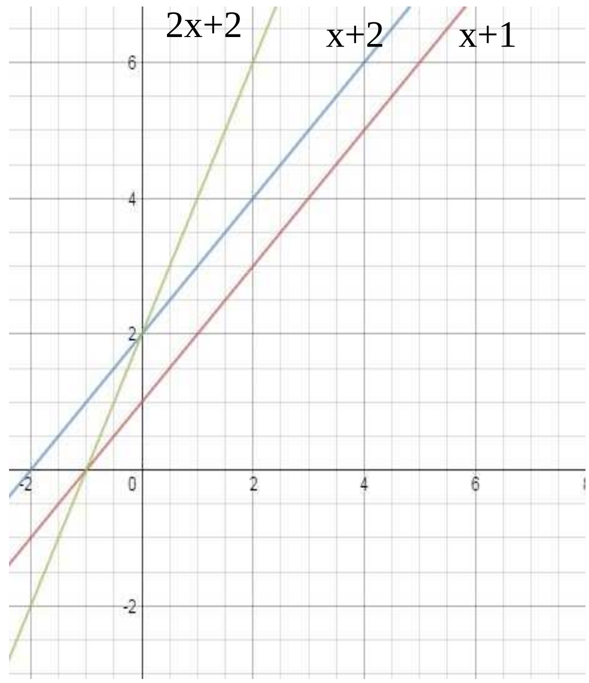


Example (Different Growth Rates) $f(x) = (x-1)^2$ and $g(x) = x$. Eventually, the graph of f lies above the graph of g, but there is no constant c for which the graph of $cg(x)$ eventually lies above that of f. We can show that, for all $c > 0$, $f(x)$ is greater than $cg(x)$ eventually. Follows from the fact that x^2 is eventually greater than dx for any positive d.

We say that f *grows faster than* g and also that f is *asymptotically greater than* g.



Example ("Greater Than" not the same as "Asymptotically Greater Than"). Consider $f(x) = x + 1$ and $g(x) = x + 2$. Certainly, g lies above f everywhere, so $f(x) < g(x)$ for all x. But g is *not* asymptotically greater since we can multiply f by 2, and then $2f(x)$ overtakes $g(x)$ for $x \geq 0$. So, asymptotically, growth rate of g is the same as growth rate of f.



4 Mathematical Induction

The idea: Suppose you wish to prove that some statement $\phi(n)$, which asserts something about each whole number n, is true for every n. For example, to prove that for all $n \geq 0$, $n < 2^n$, we would use "$n < 2^n$" as our statement $\phi(n)$. We wish to show that this statement holds for every n. Suppose now that we can prove two things:

(A) that $\phi(0)$ is true (in our example, this would mean that we can prove $0 < 2^0$);

(B) that, for any n, if $\phi(n)$ happens to be true, then $\phi(n+1)$ must also be true (in our example, this would mean that, if it happens to be true that $n < 2^n$, then it must be true that $n + 1 < 2^{n+1}$).

Mathematical Induction says that, if you can prove both (1) and (2), then you have proven that, for every n, $\phi(n)$ is indeed true.

5 Standard Induction

Suppose $\phi(n)$ is a statement depending on n. If

1. $\phi(0)$ is true, and
2. under the assumption that $n \geq 0$ and $\phi(n)$ is true, you can prove that $\phi(n+1)$ is also true,

then $\phi(n)$ holds true for all natural numbers n.

Here is a slight generalization:

General Induction Suppose $\phi(n)$ is a statement depending on n and suppose $k \geq 0$ is an integer. If

1. $\phi(k)$ is true, and
2. under the assumption that $n \geq k$ and $\phi(n)$ is true, you can prove that $\phi(n+1)$ is also true,

then $\phi(n)$ holds true for all natural numbers $n \geq k$.

In General Induction, the step in the proof where $\phi(k)$ is verified is called the *Basis Step*. The second step, where $\phi(n+1)$ is proved assuming $\phi(n)$, is called the *Induction Step*. As we reason during this second step, we will typically need to make use of $\phi(n)$ as an assumption; in this context, $\phi(n)$ is called the *induction hypothesis*.

6 Example of Mathematical Induction

We will prove that for every positive integer n,

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}.$$

We take this formula to be the formula $\phi(n)$ that we will use in the induction; that is, we let $\phi(n)$ be the statement

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}.$$

For the Basis Step, notice that $\phi(1)$ is the statement

$$\sum_{i=1}^{1} i = \frac{1(1+1)}{2}$$

which is obviously true. For the Induction Step, we assume $\phi(n)$ is true, and we prove $\phi(n+1)$. $\phi(n+1)$ is the following statement:

$$\sum_{i=1}^{n+1} i = \frac{(n+1)(n+2)}{2}$$

To prove $\phi(n+1)$ is true, we follow these steps:

$$\begin{aligned}
\sum_{i=1}^{n+1} i &= \Big(\sum_{i=1}^{n} i\Big) + (n+1) \\
&= \frac{n(n+1)}{2} + (n+1) \qquad \text{(by Induction Hypothesis)} \\
&= \frac{n(n+1)}{2} + \frac{2(n+1)}{2} \\
&= \frac{(n+1)(n+2)}{2}
\end{aligned}$$

7 The Division Algorithm

(1) Suppose m, n are positive integers. Dividing n by m gives a quotient and remainder.

(2) Example: Divide 17 by 3: Quotient is 5 and remainder is 2. Using integer division and Java mod notation, we can write:

1. quotient = 17/3
2. remainder = 17 % 3

Using mathematical notation:

1. quotient = $\lfloor 17/3 \rfloor$
2. remainder = $17 \mod 3$

We can write:

$$17 = \text{quotient} \cdot 3 + \text{remainder} = \lfloor 17/3 \rfloor \cdot 3 + 17 \bmod 3.$$

(3) In general, for any positive integers m, n, there are unique q, r so that

$$n = mq + r \qquad \text{and} \qquad 0 \leq r < m.$$

In other words

$$n = m \cdot \lfloor \frac{n}{m} \rfloor + n \bmod m.$$

8 Calculus

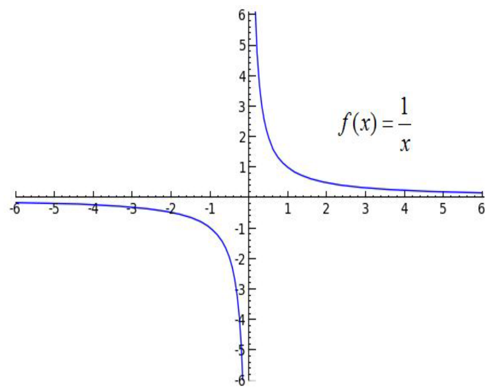
For this Algorithms course, it is not necessary to have an in-depth understanding of calculus, but it *is* important to know a few of the simple concepts and formulas, which we review here. The two concepts to be familiar with are:

(1) Limits at infinity. Example: $\lim_{n\to\infty}(n+1)/n^2 = 0$.

(2) Derivative formulas. Example: $\frac{d}{dx}(x^2 - x + 1) = 2x - 1$.

Limits at Infinity

Consider the following graph of $f(x) = \frac{1}{x}$:



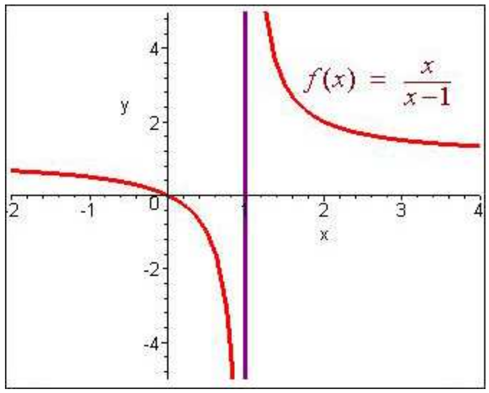
As x gets bigger and bigger, $f(x)$ gets closer and closer to 0. We write

$$\lim_{x\to\infty} \frac{1}{x} = 0.$$

Since we will be working with the set $\mathbf{N}$ of natural numbers, instead of the set $\mathbf{R}$ of real numbers, we will express this limit as

$$\lim_{n\to\infty} \frac{1}{n} = 0.$$

The following is the graph of $f(x) = \frac{x}{x-1}$:



Here, as x gets large, the graph approaches the line $y = 1$. We write:

$$\lim_{x\to\infty} \frac{x}{x-1} = 1$$

or when we are dealing only with natural numbers:

$$\lim_{n\to\infty} \frac{n}{n-1} = 1$$

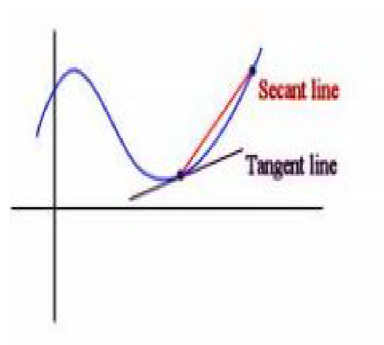
We can compute this limit algebraically by factoring from numerator and denominator the reciprocal of the highest power of x that occurs in the expression:

$$\begin{aligned}\lim_{n\to\infty} \frac{n}{n-1} &= \lim_{n\to\infty} \left(\frac{n}{n-1} \cdot \frac{1/n}{1/n}\right) \\ &= \lim_{n\to\infty} \frac{1}{1-\frac{1}{n}} \\ &= 1.\end{aligned}$$

Derivatives

The derivative of a function $f(x)$, which is written in any of these ways: $f'(x)$, $\frac{d}{dx}f(x)$, $\frac{dy}{dx}$, represents the *slope of the line tangent to the graph of f at the point* (x, y).

For example:



There are a number of convenient formulas for computing derivatives of familiar functions:

(1) $\frac{d}{dx}\, a = 0$ for any real number a.

(2) $\frac{d}{dx}\, x^r = rx^{r-1}$, for any real number $r \neq 0$.

(3) $\frac{d}{dx}\, 2^x = 2^x \,\ln 2$

(4) $\frac{d}{dx}\, \log x = \frac{1}{x} \cdot \log e$

(5) For any functions $f(x), g(x)$ (whose derivatives exist) and real numbers a, b:

   (a) (Linearity Rule) $\frac{d}{dx}(af(x) + bg(x)) = a\frac{d}{dx}\, f(x) + b\frac{d}{dx}\, g(x)$

   (b) (Product Rule) $\frac{d}{dx}(f(x) \cdot g(x)) = f(x) \cdot \frac{d}{dx}\, g(x) + g(x) \cdot \frac{d}{dx}\, f(x)$

   (c) (Reciprocal Rule) $\frac{d}{dx}(\frac{1}{f(x)}) = \frac{-f'(x)}{[f(x)]^2}$

For example:

(a) $\frac{d}{dx}\, ax = a$

(b) $\frac{d}{dx}\, ax^2 = 2ax$

(c) $\frac{d}{dx}\, ax^3 = 3ax^2$

(d) $\frac{d}{dx}\, \frac{1}{x} = -\frac{1}{x^2}$.

(e) $\frac{d}{dx}\, \sqrt{x} = \frac{d}{dx}\, x^{1/2} = \frac{1}{2}x^{-1/2} = \frac{1}{2\sqrt{x}}$.

Detecting Growth Rates Using Limits

We can tell linear functions $f(n) = an + b$ always grow more slowly than the quadratic $g(n) = n^2$ because the quotient $f(n)/g(n)$ tends to 0 as n becomes large:

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{an+b}{n^2} = \lim_{n\to\infty} \frac{\frac{a}{n} + \frac{b}{n^2}}{1} = 0.$$

Example: Show that 5n + 3 grows more slowly than n^2

Solution:

$$\lim_{n\to\infty} \frac{5n+3}{n^2} = \lim_{n\to\infty} \frac{\frac{5}{n} + \frac{3}{n^2}}{1} = 0$$

(continued)

On the other hand, all quadratic functions always grow at the same rate. We can see this using limits: If $f(n) = an^2 + bn + c$ and $g(n) = dn^2 + en + r$, where $a \neq 0$ and $d \neq 0$, then the quotient $f(n)/g(n)$ tends to a nonzero number:

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{an^2 + bn + c}{dn^2 + en + r} = \lim_{n\to\infty} \frac{a + \frac{b}{n} + \frac{c}{n^2}}{d + \frac{e}{n} + \frac{r}{n^2}} = \frac{a}{d} \neq 0.$$

Example: Show that $3n^2 + 7$ grows at the same rate as $5n^2 - n$.

Solution: $$\lim_{n\to\infty} \frac{3n^2 + 7}{5n^2 - n} = \lim_{n\to\infty} \frac{3 + \frac{7}{n^2}}{5 - \frac{1}{n}} = \frac{3}{5} \neq 0$$

Theta, Little-oh, Little-omega

Suppose f(n) and g(n) are functions. Then

- f(n) is o(g(n)) ("f(n) is little-oh of g(n)") if $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$
  [f(n) grows much more slowly than g(n)]

- f(n) is ω(g(n)) ("f(n) is little-omega of g(n)") if g(n) is o(f(n))
  [f(n) grows much faster than g(n)]

- f(n) is Θ(g(n)) ("f(n) is theta of g(n)") if for some nonzero number ***r***
  $\lim_{n\to\infty} \frac{f(n)}{g(n)} = r$ [f(n) grows at the same rate as g(n)]

The classes of functions represented by o, ω, and Θ are called complexity classes.

Note: It is theoretically possible that limits of this kind may not exist. This situation almost never arises in the context of determining running times of algorithms, and so we do not attempt to handle this special case in this course.

Examples

- $5n + 3$ is $o(n^2)$
- $3n^2 + 7$ is $\Theta(n^2)$
- n^2 is $\omega(n)$.

Polynomial Theorem

When functions are polynomials (as in the examples) there is an easy formula for determining their complexity class

- *Degree of a polynomial*. Suppose $f(n) = a_m n^m + a_{m-1} n^{m-1} + \ldots + a_1 n + a_0$. Then if $a_m \neq 0$, f(n) is called a *polynomial of degree m.* We say *the degree of f is m*.
- *Polynomial Theorem.* Suppose f(n) and g(n) are polynomials and degree of f is m, degree of g is t. Then
  - if $m = t$, then f(n) is $\Theta(g(n))$
  - if $m < t$, then f(n) is $o(g(n))$
  - if $m > t$, then f(n) is $\omega(g(n))$

Standard Complexity Classes

- The most common complexity classes used in analysis of algorithms are, in increasing order of growth rate:

  $\Theta(1), \Theta(\log n), \Theta(n^{1/k}), \Theta(n), \Theta(n\log n), \Theta(n^k)\ (k > 1),$

  $\Theta(2^n), \Theta(n!), \Theta(n^n)$

  Functions that belong to classes in the first row are known as *polynomial time bounded.*

- Verification of the relationships between these classes sometimes requires the use of *L'Hopital's Rule*

L'Hopital's Rule. Suppose f and g have derivates (at least when x is large) and their limits as $x \to \infty$ are either both 0 or both infinite. Then

$$\lim_{x\to\infty} \frac{f(x)}{g(x)} = \lim_{x\to\infty} \frac{f'(x)}{g'(x)}$$

as long as these limits exist.

Example using L'Hopital

Problem. Show that $\log n$ is $o(\sqrt{n})$.

Solution. We show that the limit of the quotient is 0.

$$\lim_{n\to\infty} \frac{\log n}{n^{1/2}} = \lim_{n\to\infty} \frac{\frac{1}{n} \cdot \log e}{(1/2)n^{-(1/2)}} = \lim_{n\to\infty} \frac{2\log e}{n^{1/2}} = 0.$$

Example Using L'Hopital

Problem. Show that $(n/2)\log n/2$ is $\Theta(n\log n)$.

Solution.

$$\begin{aligned}
\lim_{n\to\infty}\frac{(n/2)\log n/2}{n\log n} &= \lim_{n\to\infty}\frac{\log n/2}{2\log n}\\
&= \lim_{n\to\infty}\frac{\frac{1}{2}\frac{1}{\frac{n}{2}}\cdot\log e}{\frac{2\log e}{n}}\\
&= \frac{1}{2}.
\end{aligned}$$

Big-oh and Big-omega

- If f(n) *grows no faster* than g(n), we say f(n) is O(g(n)) ("big-oh")
- If f(n) *grows at least as fast* as g(n), we say f(n) is Ω(g(n)) ("big-omega")
- Limit criterion:

f(n) is O(g(n)) if

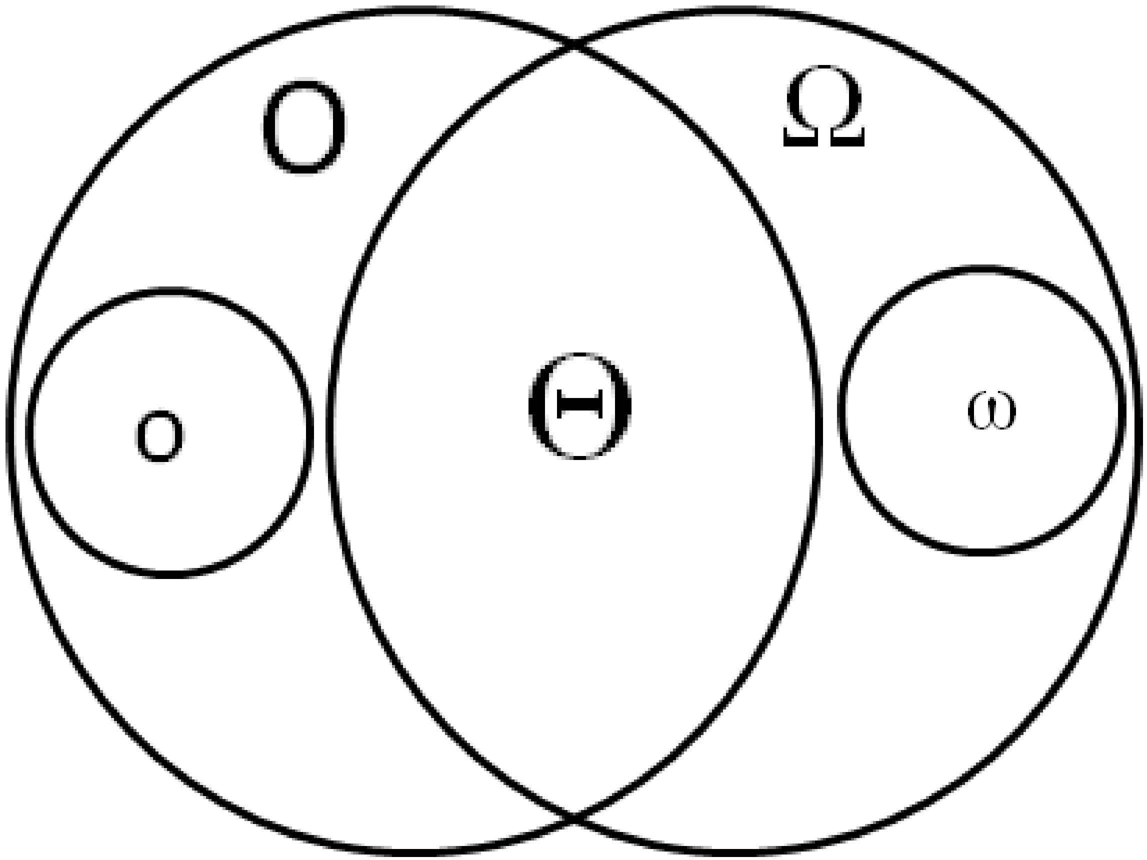
$$\text{There is } r \geq 0 \text{ such that } \lim_{n \to \infty} \frac{f(n)}{g(n)} = r.$$

Then, f(n) is Ω(g(n)) if g(n) is O(f(n)).

Examples

- Both $2n + 1$ and $3n^2$ are $O(n^2)$
- Both $2n^2 - 1$ and $4n^3$ are $\Omega(n^2)$

Relationships Between the Complexity Classes



- Whenever f(n) is o(g(n)), f(n) is O(g(n).
- Whenever f(n) is ω(g(n)), f(n) is Ω(g(n)).
- No function is in both o and ω
- If f(n) is in both O(g(n)) and Ω(g(n)), it is in Θ(g(n)).

Summary of Criteria for Determining Complexity

$f(n)$ is $O(g(n))$	*if*	$\lim\limits_{n\to\infty} \frac{f(n)}{g(n)}$ is finite
$f(n)$ is $\Omega(g(n))$	*if*	$\lim\limits_{n\to\infty} \frac{g(n)}{f(n)}$ is finite
$f(n)$ is $\Theta(g(n))$	*if*	$\lim\limits_{n\to\infty} \frac{f(n)}{g(n)}$ is nonzero
$f(n)$ is $o(g(n))$	*if*	$\lim\limits_{n\to\infty} \frac{f(n)}{g(n)} = 0$
$f(n)$ is $\omega(g(n))$	*if*	$\lim\limits_{n\to\infty} \frac{g(n)}{f(n)} = 0$