Math Facts

Asymptotic notation

Motivation:

Suppose that on inputs of size n, Algorithm 1 takes time

$$f(n) = n^2 + 10n$$

while Algorithm 2 takes time

$$g(n) = 10n \log_2 n + 100n$$

Since

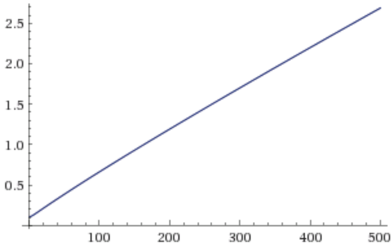
$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = \infty$$

Eventually, as n gets larger, Algorithm 1 will get slower and slower than Algorithm 2

Input interpretation:

plot	$\dfrac{n^2 + 10\,n}{10\,n \log_2(n) + 100\,n}$	$n = 1$ to 500

Plot:



In this class, we are mainly concerned with the *rate of growth* of the running time as a function of the *input size*

For this purpose, we really only need to worry about the "high order term" of this function

We typically ignore the leading constant

- For $f(n) = n^2 + 10n$, we just say $f(n) = O(n^2)$
- For $g(n) = 10n\log_2 n + 100n$, we just say $g(n) = O(n\log_2 n)$
- Since $\log_b n = \log_c n / \log_c b$, we can just say $g(n) = O(n\log n)$ — the base of the logarithm doesn't really matter *(as long as it is a* constant*)*

The key idea: for large enough n, an $O(n\log n)$ algorithm will be faster than an $O(n^2)$ algorithm

Some formal definitions

Definition: Let f and g be functions from $\mathbb{Z}_{>0}$ to $\mathbb{R}$.

We say $f = O(g)$ if

$$|f(n)| \leq c|g(n)|$$

for some constant c and all sufficiently large n

Or more precisely:

$$\exists c \in \mathbb{R}_{>0}\ \exists n_0 \in \mathbb{Z}_{>0}\ \forall n \geq n_0 : \quad |f(n)| \leq c|g(n)|$$

Intuition: $f = O(g)$ means f grows *no faster* than g

The relation "$f = O(g)$" is analogous to "$f \leq g$"

Example: $g(n) = 10n\log_2 n + 100n$

Claim: $g(n) = O(n\log_2 n)$

$$\lim_{n\to\infty} \frac{g(n)}{n\log_2 n} = \lim_{n\to\infty} \frac{10n\log_2 n + 100n}{n\log_2 n}$$

$$= \lim_{n\to\infty} \frac{10 + 100/\log_2 n}{1} = \frac{10+0}{1} = 10$$

By the definition of a limit: for all $\epsilon > 0$,

$$10 - \epsilon \le \frac{g(n)}{n\log_2 n} \le 10 + \epsilon$$

for all sufficiently large n

In particular, setting $\epsilon := 0.1$:

$$g(n) \le 10.1 \cdot n\log_2 n$$

for all sufficiently large n

Implicit big-O notation

If $f(n) = 2n^2 + 10n + 1$, we may also write

$$f(n) = 2n^2 + O(n)$$

This means that

$$f(n) = 2n^2 + h(n)$$

for some function $h = O(n)$

Useful in situations where we do not want to completely ignore the constant in the high-order term

Big-Omega notation

$f = \Omega(g)$ means $g = O(f)$

The relation "$f = \Omega(g)$" is analogous to "$f \geq g$"

big-Theta notation

Definition: Let f and g be functions from $\mathbb{Z}_{>0}$ to $\mathbb{R}$.

We say $f = \Theta(g)$ if $f = O(g)$ and $g = O(f)$

Equivalently:

$$|f(n)| \leq c|g(n)| \quad \text{and} \quad |f(n)| \geq d|g(n)|$$

for some constants c, d and all sufficiently large n

Intuition: $f = \Theta(g)$ means f and g grow *at the same rate*

The relation "$f = \Theta(g)$" is analogous to "$f = g$"

Note: $f = \Theta(g)$ is a *symmetric relation*:

$$f = \Theta(g) \iff g = \Theta(f)$$

Example: $f(n) = n^2 + 10n$, $g(n) = 2n^2 + n$

Claim: $f = \Theta(g)$

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{n^2 + 10n}{2n^2 + n} = \lim_{n\to\infty} \frac{1 + 10/n}{2 + 1/n} = \frac{1+0}{2+0} = \frac{1}{2}$$

$\therefore$ $f(n) \leq 0.75 \cdot g(n)$ and $f(n) \geq 0.25 \cdot g(n)$ for all sufficiently large n

little-o notation

Definition: Let f and g be functions from $\mathbb{Z}_{>0}$ to $\mathbb{R}$.

We say $f = o(g)$ if

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$$

Equivalently: for every $\epsilon \in \mathbb{R}_{>0}$

$$|f(n)| < \epsilon |g(n)|$$

for all sufficiently large n

Intuition: $f = o(g)$ means f grows <u>*strictly*</u> *more slowly* than g

The relation "$f = o(g)$" is analogous to "$f < g$"

Example: $f(n) = 2n^2 + 10n$, $g(n) = n^3 + n$

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{2n^2 + 10n}{n^3 + n} = \lim_{n\to\infty} \frac{2/n + 10/n^2}{1 + 1/n^2}$$
$$= \frac{0+0}{1+0} = 0$$

$\therefore f = o(g)$

General fact: if $f = o(g)$, then

- $f = O(g)$
- $g \neq O(f)$

Implicit little-o notation

$f(n) = 2n^2 + 10n = 2n^2 + o(n^2) = 2n^2(1 + o(1))$

$f(n) = 2^{n+1/n} = 2^{n+o(1)}$

Example: $f(n) = n^\alpha$, $g(n) = n^\beta$, where $\alpha < \beta$ are constants

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{n^\alpha}{n^\beta} = \lim_{n\to\infty} n^{\alpha-\beta} = 0 \qquad \therefore f = o(g)$$

Example: $f(n) = \ln(n)$, $g(n) = n^\beta$, where $\beta > 0$ is a constant

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{\ln(n)}{n^\beta} = \lim_{n\to\infty} \frac{n^{-1}}{\beta n^{\beta-1}} \quad \text{(L'hospital's rule)}$$

$$= \lim_{n\to\infty} \frac{1}{\beta n^\beta} = 0 \qquad \therefore f = o(g)$$

Example: $f(n) = \ln(n)^\alpha$, $g(n) = n^\beta$, where α, β are positive constants

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{\ln(n)^\alpha}{n^\beta}$$

$$= \lim_{n\to\infty} \left(\frac{\ln(n)}{n^{\beta/\alpha}}\right)^\alpha = \left(\lim_{n\to\infty} \frac{\ln(n)}{n^{\beta/\alpha}}\right)^\alpha = 0^\alpha = 0 \qquad \therefore f = o(g)$$

"any power of $\log n$ *grows more slowly than any power of* n*"*

Limit Comparison Theorem for Rates of Growth: *general tool for comparing growth rates of functions*

Let f and g be functions from $\mathbb{Z}_{>0}$ to $\mathbb{R}$.

Suppose

$$0 \leq L := \lim_{n \to \infty} \frac{|f(n)|}{|g(n)|} \leq \infty$$

Then we have:

- If $L = 0$ then $f = o(g)$
- If $L = \infty$ then $g = o(f)$
- If $0 < L < \infty$ then $f = \Theta(g)$

A more concrete tool

Suppose f is a sum of terms, where each term of the form $cn^{\alpha}(\log n)^{\beta}$, for constants c, α, β, where $c \neq 0$

We can sort the terms:

- $cn^{\alpha}(\log n)^{\beta}$ is a *higher term* than $c'n^{\alpha'}(\log n)^{\beta'}$ if
    - $\alpha > \alpha'$, or
    - $\alpha = \alpha'$ and $\beta > \beta'$

For two such functions f and g, we can compare their highest terms:

- If g's highest term is higher than f's: $f = o(g)$
- If f's highest term is higher than g's: $g = o(f)$
- Otherwise: $f = \Theta(g)$

Pop Quiz!

$f(n)$	$g(n)$	$f = o(g)$?	$g = o(f)$?	$f = \Theta(g)$?
$3n^2$	$2n^3$			
$2n^3 + 10n^2$	$n^2 + n$			
$n^2 \log_2 n + 10n^2$	$n^3 - n$			
$n^2 / \log_2 n$	$n \log_2 n$			
$(\log_2 n)^2$	$\sqrt{n}$			
$\log_5 n$	$\log_3 n$			
5^n	3^n			

Counting steps

for i in $[1..n]$ do
    for j in $[1..i]$ do
        HERE

How many times does line "HERE" get executed?

For each iteration of the outer for loop, it gets executed i times:

$$S = \sum_{i=1}^{n} i = n(n+1)/2 = n^2/2 + O(n) = \Theta(n^2)$$

The tale of young Gauss and the sum of the first 100 positive integers

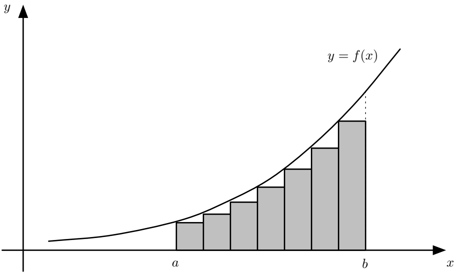
Old Gauss (1777–1855) on German banknote

$$
\begin{array}{rrcrcccrcr}
S = & 1 & + & 2 & + & \cdots & + & 99 & + & 100 \\
S = & 100 & + & 99 & + & \cdots & + & 2 & + & 1 \\
\hline
2S = & 101 & + & 101 & + & \cdots & + & 101 & + & 101 \\
2S = & 100 \cdot 101 & & & & & & & &
\end{array}
$$

What about more complicated sums? $\sum_i i^2$, $\sum_i 1/i$

Approximating sums by integrals (1)

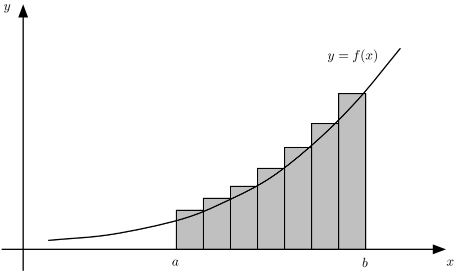
Suppose f is continuous and *non-decreasing* on $[a, b]$, where $a, b \in \mathbb{Z}$



$$\sum_{i=a}^{b-1} f(i) \leq \int_a^b f(x)dx$$

Approximating sums by integrals (2)

Suppose f is continuous and *non-decreasing* on $[a, b]$, where $a, b \in \mathbb{Z}$



$$\int_a^b f(x)dx \leq \sum_{i=a+1}^{b} f(i)$$

Approximating sums by integrals (3)

More general statement

Suppose f is continuous and *monotone* (non-decreasing or non-increasing) on $[a, b]$, where $a, b \in \mathbb{Z}$

Then

$$\int_a^b f(x)dx + m \;\leq\; \sum_{i=a}^{b} f(i) \;\leq\; \int_a^b f(x)dx + M,$$

where $m := \min(f(a), f(b))$, and $M := \max(f(a), f(b))$

Example: Estimate $Q_n := \sum_{i=1}^{n} i^2$

$$\int_1^n x^2 dx = \left[x^3/3 \right]_1^n = n^3/3 - 1/3$$

Therefore: $n^3/3 + 2/3 \le Q_n \le n^3/3 + n^2 - 1/3$

$Q_n = n^3/3 + O(n^2)$

Example: Estimate $H_n := \sum_{i=1}^{n} 1/i$

$$\int_1^n (1/x) dx = \left[\ln(x) \right]_1^n = \ln(n)$$

Therefore: $\ln(n) + 1/n \le H_n \le \ln(n) + 1$

$H_n = \ln(n) + O(1)$

H_n is called the nth *Harmonic Number*

Geometric series

$$S_n(r) := \sum_{i=0}^{n} r^i = 1 + r + r^2 + \cdots + r^n$$

Fact: for $r \neq 1$, we have

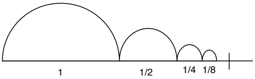
$$S_n(r) = \frac{1 - r^{n+1}}{1 - r}$$

Proof:

$$(1-r)S_n(r) = (1 + r + \cdots + r^n) - (r + r^2 + \cdots r^{n+1}) = 1 - r^{n+1}$$

Example: $S_n(2) = 1 + 2 + 4 + \cdots + 2^n = 2^{n+1} - 1$

Example: $S_n(1/2) = 1 + 1/2 + 1/4 + \cdots + 1/2^n = 2 - 1/2^n$



Infinite Series

$$\sum_{i=0}^{\infty} a_i = \lim_{n\to\infty} \sum_{i=0}^{n} a_i$$

Example:

$$\begin{aligned} 1 + 1/2 + 1/4 + \cdots &= \sum_{i=0}^{\infty} 1/2^i = \lim_{n\to\infty} \sum_{i=0}^{n} 1/2^i \\ &= \lim_{n\to\infty} (2 - 1/2^n) = 2 \end{aligned}$$

More generally, for $|r| < 1$:

$$\sum_{i=0}^{\infty} r^i = \lim_{n\to\infty} \sum_{i=0}^{n} r^i = \lim_{n\to\infty} (1 - r^{n+1})/(1 - r) = 1/(1 - r)$$

Consider $\sum_{i=0}^{\infty} a_i$, where $a_i \geq 0$

Fact: either $\sum_{i=0}^{\infty} a_i$ *converges* to a constant, or *diverges* to ∞

Clearly, if $\sum_{i=0}^{\infty} a_i < \infty$, then $a_i \longrightarrow 0$

But the converse is not true!

It may the case that $\sum_{i=0}^{\infty} a_i = \infty$, even if $a_i \longrightarrow 0$

Example:

$$\sum_{i=1}^{\infty}(1/i) = \lim_{n\to\infty}\sum_{i=1}^{n}(1/i) = \lim_{n\to\infty}\Big(\ln(n) + O(1)\Big) = \infty$$

The ratio test

Consider the series $\sum_{i=0}^{\infty} a_i$, where $a_i \geq 0$

Let $L := \lim_{i\to\infty} a_{i+1}/a_i$

- If $L < 1$, the series converges
- If $L > 1$, the series diverges to infinity
- If $L = 1$ (or the limit does not exist), the test is *inconclusive*

Example:

$$\sum_{i=0}^{\infty} \frac{i}{2^i}: \qquad \frac{a_{i+1}}{a_i} = \frac{i+1}{2i} \longrightarrow 1/2 \qquad \therefore \text{ converges}$$

Example:

$$\sum_{i=1}^{\infty} \frac{1}{i}: \qquad \frac{a_{i+1}}{a_i} = \frac{i}{i+1} \longrightarrow 1 \qquad \therefore \text{ inconclusive}$$