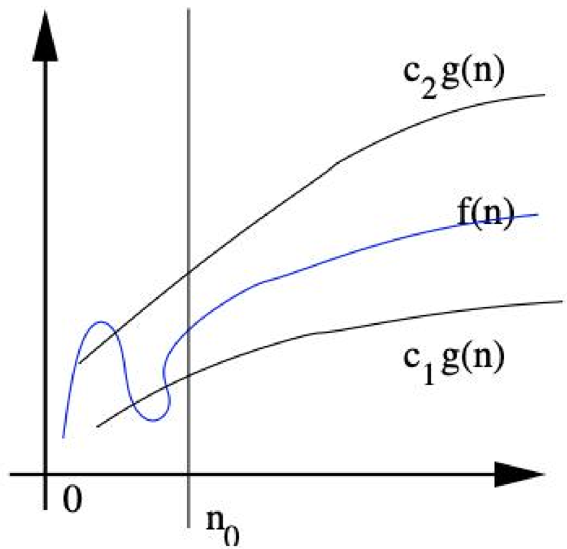
Growth rate functions

We want to define the precise meaning of growth rate.

Definition 1:

$$\Theta(g(n)) = \{f(n) \quad | \quad \exists c_1 > 0, c_2 > 0, n_0 \geq 0 \text{ so that } \\ \forall n \geq n_0, 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)\}$$

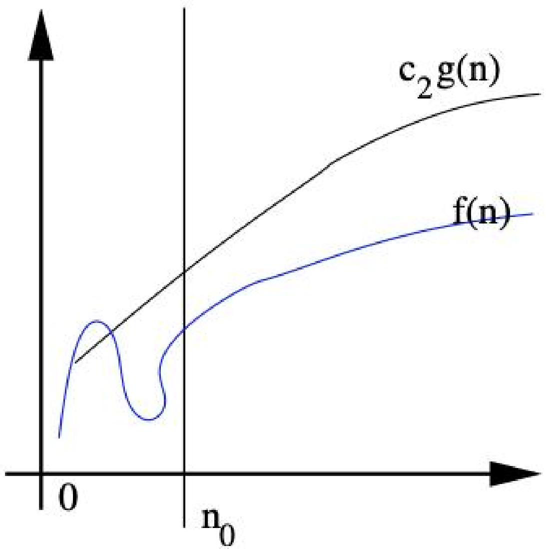
If $f(n) \in \Theta(g(n))$, we also write $f(n) = \Theta(g(n))$ and say: the growth rate of $f(n)$ is the same as the growth rate of $g(n)$.



Definition 2:

$$O(g(n)) = \{f(n) \quad | \quad \exists c_2 > 0, n_0 \geq 0 \text{ so that} \\ \forall n \geq n_0, 0 \leq f(n) \leq c_2 g(n)\}$$

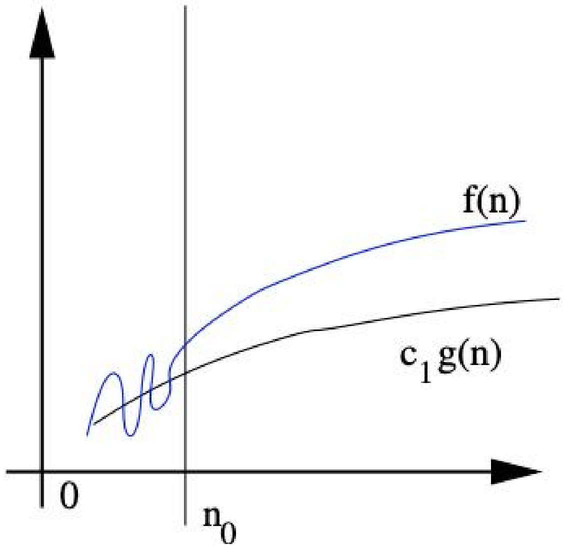
If $f(n) \in O(g(n))$, we also write $f(n) = O(g(n))$ and say: the growth rate of $f(n)$ is at most the growth rate of $g(n)$.



Definition 3:

$$\Omega(g(n)) = \{f(n) \quad | \quad \exists c_1 > 0, n_0 \geq 0 \text{ so that } \\ \forall n \geq n_0, 0 \leq c_1 g(n) \leq f(n)\}$$

If $f(n) \in \Omega(g(n))$, we also write $f(n) = \Omega(g(n))$ and say: the growth rate of $f(n)$ is at least the growth rate of $g(n)$.



Definition 4:

$$o(g(n)) = \{f(n) \quad | \quad \forall c > 0, \exists n_0 \geq 0 \text{ so that}$$
$$\forall n \geq n_0, 0 \leq f(n) \leq cg(n)\}$$

If $f(n) \in o(g(n))$, we also write $f(n) = o(g(n))$ and say: the growth rate of $f(n)$ is strictly less than the growth rate of $g(n)$.

Definition 5:

$$\omega(g(n)) = \{f(n) \quad | \quad \forall c > 0, \exists n_0 \geq 0 \text{ so that}$$
$$\forall n \geq n_0, 0 \leq cg(n) \leq f(n)\}$$

If $f(n) \in \omega(g(n))$, we also write $f(n) = \omega(g(n))$ and say: the growth rate of $f(n)$ is strictly bigger than the growth rate of $g(n)$.

The properties of growth rate functions:

The meaning of these notations (roughly speaking):

if	the growth-rate is
$f(n) = \Theta(g(n))$	$=$
$f(n) = O(g(n))$	$\leq$
$f(n) = \Omega(g(n))$	$\geq$
$f(n) = o(g(n))$	$<$
$f(n) = \omega(g(n))$	$>$

The properties of growth rate functions:

Some properties of growth rate functions:

- $f(n) = \Theta(g(n)) \iff f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$
- $f(n) = O(g(n)) \iff g(n) = \Omega(f(n))$

The properties of growth rate functions:

Some properties of growth rate functions:

- $f(n) = \Theta(g(n)) \iff f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$
- $f(n) = O(g(n)) \iff g(n) = \Omega(f(n))$
- $f(n) = o(g(n)) \iff g(n) = \omega(f(n))$

The properties of growth rate functions:

Some properties of growth rate functions:

- $f(n) = \Theta(g(n)) \Longleftrightarrow f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$
- $f(n) = O(g(n)) \Longleftrightarrow g(n) = \Omega(f(n))$
- $f(n) = o(g(n)) \Longleftrightarrow g(n) = \omega(f(n))$
- $f(n) = O(g(n))$ and $g(n) = O(h(n)) \Longrightarrow f(n) = O(h(n))$ if we replace O by $\Theta, \Omega, o, \omega$, it holds true.

The properties of growth rate functions:

Some properties of growth rate functions:

- $f(n) = \Theta(g(n)) \Longleftrightarrow f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$
- $f(n) = O(g(n)) \Longleftrightarrow g(n) = \Omega(f(n))$
- $f(n) = o(g(n)) \Longleftrightarrow g(n) = \omega(f(n))$

- $f(n) = o(g(n)) \iff g(n) = \omega(f(n))$
- $f(n) = O(g(n))$ and $g(n) = O(h(n)) \Longrightarrow f(n) = O(h(n))$ if we replace O by $\Theta, \Omega, o, \omega$, it holds true.
- $f(n) = o(g(n)) \Longrightarrow f(n) = O(g(n))$
- $f(n) = \omega(g(n)) \Longrightarrow f(n) = \Omega(g(n))$
- Read Ch. 3 for more relations and properties.

The properties of growth rate functions:

As an example, we prove one property:

$$f(n) = O(g(n)) \text{ and } g(n) = O(h(n)) \Longrightarrow f(n) = O(h(n)) \qquad (1)$$

Proof: Suppose $f(n) = O(g(n))$. By definition, this means:

$$\exists c_1 > 0, n_1 \geq 0 \text{ so that } \forall n \geq n_1, 0 \leq f(n) \leq c_1 g(n) \qquad (2)$$

Similarly, $g(n) = O(h(n))$ means:

$$\exists c_2 > 0, n_2 \geq 0 \text{ so that } \forall n \geq n_2, 0 \leq g(n) \leq c_2 h(n) \qquad (3)$$

Take $n_0 = \max\{n_1, n_2\}$. Then for all $n \geq n_0$ both (1) and (2) are true. Hence: $f(n) \leq c_1 \cdot g(n) \leq c_1 \cdot c_2 \cdot h(n) = c \cdot h(n)$, where $c = c_1 \cdot c_2$. By the definition of O, this means $f(n) = O(h(n))$. Thus the property (1) holds.

Limit Test

Limit Test is a powerful method for comparing functions.

Limit Test

Let $T_1(n)$ and $T_2(n)$ be two functions. Let $c = \lim_{n\to\infty} \frac{T_1(n)}{T_2(n)}$.

1. If c is a constant > 0, then $T_1(n) = \Theta(T_2(n))$.
2. If $c = 0$, then $T_1(n) = o(T_2(n))$.
3. If $c = \infty$, then $T_1(n) = \omega(T_2(n))$.
4. If c does not exists (or if we do not know how to compute c), the limit test fails.

Proof of (1): $c = \lim_{n\to\infty} \frac{T_1(n)}{T_2(n)}$ means: $\forall \epsilon > 0$, there exists $n_0 \geq 0$ such that for any $n \geq n_0$: $\left|\frac{T_1(n)}{T_2(n)} - c\right| \leq \epsilon$; or equivalently: $c - \epsilon \leq \frac{T_1(n)}{T_2(n)} \leq c + \epsilon$. Let $\epsilon = c/2$ and let $c_1 = c - \epsilon = c/2$ and $c_2 = c + \epsilon = 3c/2$, we have

$$c_1 T_2(n) \leq T_1(n) \leq c_2 T_2(n)$$

for all $n \geq n_0$. Thus $T_1(n) = \Theta(T_2(n))$ by definition.

L'Hospital Rule

L'Hospital Rule

- If $\lim_{n\to\infty} f(n) = 0$ and $\lim_{n\to\infty} g(n) = 0$, then

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

- If $\lim_{n\to\infty} f(n) = \infty$ and $\lim_{n\to\infty} g(n) = \infty$, then

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

Example

Example 2

$T_1(n) = n^2 + 6$, $T_2(n) = n \lg n$. (Recall: $\lg n = \log_2 n$.)

$$\begin{aligned} \lim_{n\to\infty} \frac{T_1(n)}{T_2(n)} &= \lim_{n\to\infty} \frac{n^2+6}{n\lg n} = \lim_{n\to\infty} \frac{n+\frac{6}{n}}{\lg n} \\ &= \lim_{n\to\infty} \frac{1-\frac{6}{n^2}}{\frac{1}{\ln 2\cdot n}} \text{ (by L'Hospital Rule)} \\ &= \ln 2 \lim_{n\to\infty}(n - \tfrac{6}{n}) = \ln 2(\infty - 0) = \infty \end{aligned}$$

By Limit Test, we have $n^2 + 6 = \omega(n \lg n)$.

Stirling Formula

Stirling Formula

$$\sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{\frac{1}{12n+1}} \leq n! \leq \sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{\frac{1}{12n}}$$

or:

$$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n \cdot \left(1 + \Theta\left(\frac{1}{n}\right)\right)$$

When $n = 10$;

- $n! = 3628800$.
- $\sqrt{2\pi n}\left(\frac{n}{e}\right)^n = 3598696$, 99% accurate.

Summation formulas

$$\sum_{i=1}^{n} i^1 \quad = \quad 1 + 2 \cdots + n = \frac{n(n+1)}{2} = \Theta(n^2) \tag{3}$$

$$\sum_{i=1}^{n} i^2 \quad = \quad 1^2 + 2^2 \cdots + n^2 = \frac{n(n+1)(2n+1)}{6} = \Theta(n^3) \tag{4}$$

$$\sum_{i=1}^{n} i^3 \quad = \quad 1^3 + 2^3 \cdots + n^3 = \frac{n^2(n+1)^2}{4} = \Theta(n^4) \tag{5}$$

In general, for any $k > 0$, the following is true.

$$\sum_{i=1}^{n} i^k \quad = \quad \Theta(n^{k+1}) \tag{6}$$

Calculate $D(n)$

To calculate the second sum, let $t = j - i + 1$. When $j = i$, $t = 1$. When $j = n$, $t = n - i + 1$. Thus

$$\sum_{j=i}^{n}(j - i + 1) = \sum_{t=1}^{n-i+1} t = 1 + 2 \cdots + (n - i + 1) = \frac{(n - i + 2)(n - i + 1)}{2}$$

Next we calculate: $\sum_{i=1}^{n} \frac{(n-i+2)(n-i+1)}{2}$. Let $s = n - i + 1$. When $i = 1$, $s = n$. When $i = n$, $s = 1$. Thus:

$$\begin{aligned} \sum &= \sum_{s=1}^{n} \frac{(s+1)s}{2} = \frac{1}{2}\{\sum_{s=1}^{n} s^2 + \sum_{s=1}^{n} s\} \\ &= \frac{1}{2}\{\frac{n(n+1)(2n+1)}{6} + \frac{n(n+1)}{2}\} = \Theta(n^3) \end{aligned}$$

How to Find Summation Formula?

Fact:

For any **positive integer** k, $\sum_{i=1}^{n} i^k$ is a polynomial in n of degree $k+1$.

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

We know $\sum_{i=1}^{n} i^2$ is a polynomial of degree 3. Namely, for some unknown constants a, b, c, d, we have $\sum_{i=1}^{n} i^2 = an^3 + bn^2 + cn + d$. We need to find the values of a, b, c, d. Plug in $n = 0, 1, 2, 3$:

$$
\begin{array}{llllllllllll}
n=0: & \sum_{i=1}^{0} i^2 & = & & = & 0 & = & & & & & d \\
n=1: & \sum_{i=1}^{1} i^2 & = & 1^2 & = & 1 & = & a & + & b & + & c & + & d \\
n=2: & \sum_{i=1}^{2} i^2 & = & 1^2+2^2 & = & 5 & = & 8a & + & 4b & + & 2c & + & d \\
n=3: & \sum_{i=1}^{3} i^2 & = & 1^2+2^2+3^2 & = & 14 & = & 27a & + & 9b & + & 3c & + & d
\end{array}
$$

Solving these 4 equations, we get: $a = 1/3,\ b = 1/2\ , c = 1/6\ , d = 0$. So:

$\sum_{i=1}^{n} i^2 = \frac{n^3}{3} + \frac{n^2}{2} + \frac{n}{6} = \frac{1}{6}n(2n^2+3n+1) = \frac{1}{6}n(2n+1)(n+1)$

Exercise: Find the sum formula of $\sum_{i=1}^{n} i^3$

More Summations:

The following summation formulas are useful.

$$\sum_{i=0}^{n} a^i = 1 + a + a^2 + \cdots + a^n = \begin{cases} \frac{1-a^{n+1}}{1-a} & = \Theta(1) & \text{if } 0 < a < 1 \\ n+1 & = \Theta(n) & \text{if } a = 1 \\ \frac{a^{n+1}-1}{a-1} & = \Theta(a^n) & \text{if } 1 < a \end{cases} \tag{7}$$

$\sum_{i=0}^{n} a^i$ is called geometric series.

$$H(n) = 1 + 1/2 + 1/3 + \cdots + 1/n = \sum_{i=1}^{n} \frac{1}{i} = \Theta(\ln n) \tag{8}$$

$H(n)$ is called Harmonic series.
How to determine the growth rate of $H(n)$?

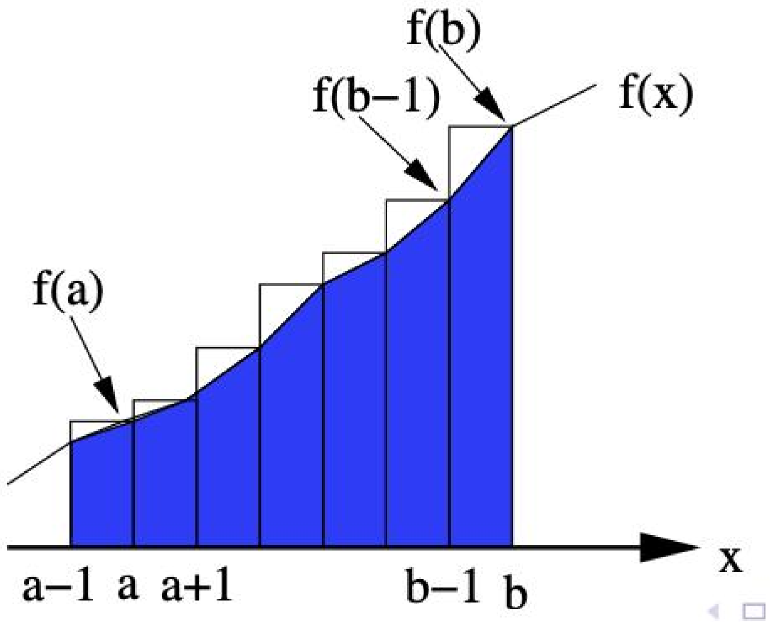
Integration Method

Integration Method

Let $f(x)$ be an increasing function. Then for any $a \leq b$:

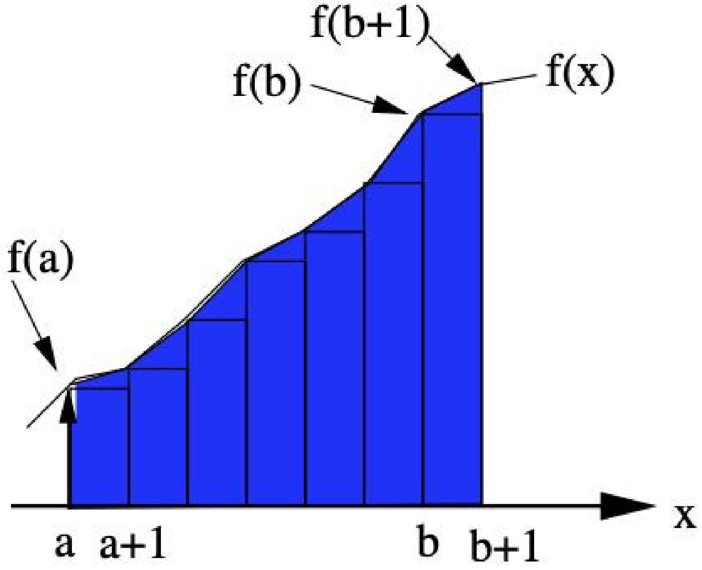
$$\int_{a-1}^{b} f(x)dx \leq \sum_{i=a}^{b} f(i) \leq \int_{a}^{b+1} f(x)dx$$

In the Fig, $\sum$ = the area of the staircase region. The first $\int$ = the area of the shaded region. Since $f(x)$ is increasing, the first $\leq$ holds.



Integration Method

In the Fig, $\sum$ is the area of the staircase region. The second $\int$ is the area of the shaded region. Since $f(x)$ is increasing, the second $\leq$ holds.



Similarly:

Let $f(x)$ be a decreasing function. Then for any $a \leq b$:

$$\int_{a-1}^{b} f(x)dx \geq \sum_{i=a}^{b} f(i) \geq \int_{a}^{b+1} f(x)dx$$

Solving Linear Recursive Equations

Fibonacci number

$$\text{Fib}_0 = 0,\ \text{Fib}_1 = 1, \ldots,\ \text{Fib}_{n+2} = \text{Fib}_{n+1} + \text{Fib}_n$$

We can compute Fib_n by above recursive definition. How to calculate it directly from n?

$$\text{Fib}_n = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n$$

- $\frac{1\pm\sqrt{5}}{2}$ are the two roots of the equation: $x^2 = x + 1$.
- Since $\left|\frac{1-\sqrt{5}}{2}\right| < 1$, the second term $\to 0$. So $\text{Fib}_n \approx \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n$.
  ($\alpha = \frac{1+\sqrt{5}}{2} = 1.618...$ is called the golden ratio.)
- For $n = 8$, $\text{Fib}_8 = 21$ where $\frac{1}{\sqrt{5}}\alpha^8 = 21.0095$.

Linear recursive sequences

Linear recursive sequences

A sequence $\{f_0, f_1, \ldots, f_n \ldots\}$ is called a linear recursive sequence of order k if it is defined as follows:

- $f_0, f_1, \ldots, f_{k-1}$ are given.
- For all $n \geq 0$, $f_{n+k} = c_{k-1} f_{n+k-1} + c_{k-2} f_{n+k-2} + \cdots + c_1 f_{n+1} + c_0 f_n$ where $c_{k-1}, c_{k-2}, \ldots c_0$ are fixed constants.

Example 1: $\{\text{Fib}_n\}$ is a linear recursive sequence of order 2 where $c_1 = 1$ and $c_0 = 1$.

Example 2: $f_0 = 1, f_1 = 2, f_2 = 4$ and for all $n \geq 0$, $f_{n+3} = 3f_{n+1} - 2f_n$
Then $\{f_n\}$ is a linear recursive sequence of order 3 where $c_2 = 0$, $c_1 = 3$ and $c_0 = -2$.

Solving linear recursive sequences

- The characteristic equation of the linear recursive seq is:

$$x^k = c_{k-1}x^{k-1} + c_{k-2}x^{k-2} + \cdots + c_1 x^1 + c_0$$

- Solve this equation for x. Let $\alpha_1, \ldots, \alpha_k$ be the roots.
- Assuming all roots are distinct. Then the solution of f_n has the form

$$f_n = a_1(\alpha_1)^n + a_2(\alpha_2)^n + \cdots + a_k(\alpha_k)^n$$

  for some constants $a_1, a_2, \ldots, a_k$.
- Plug in the initial values $f_0, f_1, \ldots, f_{k-1}$, we get k equations. Solve them to find $a_1, a_2, \ldots, a_k$.

Master Theorem

For DaC algorithms, the runtime function often satisfies:

$$T(n) = \begin{cases} O(1) & \text{if } n \leq n_0 \\ aT(n/b) + \Theta(f(n)) & \text{if } n > n_0 \end{cases}$$

- If $n \leq n_0$ (n_0 is a small constant), we solve the problem directly without recursive calls. Since the input size is fixed (bounded by n_0), it takes $O(1)$ time.
- We make a recursive calls. The input size of each is n/b. Hence the runtime $T(n/b)$.
- $\Theta(f(n))$ is the time needed by all other processing.
- $T(n) =?$

Master Theorem

Master Theorem (Theorem 4.1, Cormen's book.)

1. If $f(n) = O(n^{\log_b a - \epsilon})$ for some constant $\epsilon > 0$, then $T(n) = \Theta(n^{\log_b a})$.
2. If $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta(n^{\log_b a} \log n)$.
3. If $f(n) = \Omega(n^{\log_b a + \epsilon})$ for some constant $\epsilon > 0$, and $af(n/b) \leq cf(n)$ for some $c < 1$ for sufficiently large n, then $T(n) = \Theta(f(n))$.

Example: MergeSort

We have $a = 2, b = 2$, hence $\log_b a = \log_2 2 = 1$. So $f(n) = \Theta(n^1) = \Theta(n^{\log_b a})$.

By statement (2), $T(n) = \Theta(n^{\log_2 2} \log n) = \Theta(n \log n)$.