9.10

Big O notation

Def 1.1 Let $f, g: \mathbb{R} \to \mathbb{R}$ or $\mathbb{N} \to \mathbb{R}$, Write $f = O(g)$ if

$$(\exists c>0)\underbrace{(\exists N)(\forall x \geq N)}_{\text{when } x \text{ is large enough}}(|f(x)| \leq c \cdot |g(x)|)$$

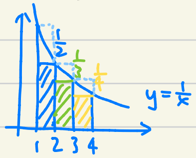
eg. $T(n) = 6n^4 - 2n^3 + 5 = O(n^4)$

$$|T(n)| = |6n^4 - 2n^3 + 5| \leq 6n^4 + 2n^3 + 5$$
$$\leq 6n^4 + 2n^4 + 5n^4, \quad n \geq 1$$
$$= 13n^4.$$

$c = 13$. $N = 1$ So, $T(n) = O(n^4)$

Think geometrically, prove algebraically.
John Tate

$1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} = \ln n + O(1)$ constant



$$1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} - \ln n$$
$$= \qquad\qquad - \int_1^n \frac{1}{x}\,dx$$
$$= 1 + \sum_{i=1}^{n-1} \frac{1}{i+1} - \sum_{i=1}^{n-1} \int_i^{i+1} \frac{1}{x}\,dx$$
$$= 1 + \sum_{i=1}^{n-1} \left(\frac{1}{i+1} - \int_i^{i+1} \frac{1}{x}\,dx\right) < 1 = O(1)$$

Prop 1.2 1) If $f_1 = O(g_1)$ and $f_2 = O(g_2)$, then $f_1 f_2 = O(g_1 g_2)$

2) $f \cdot O(g) = O(f \cdot g)$

Examples.

1) $5 + 0.001n^3 + 0.25n$ $\quad O(n^3)$

2) $500n + 100n^{1.5} + 50\, n \log_{10} n$ $\quad O(n^{1.5})$

3) $n^2 \log_2 n + n(\log_2 n)^2$ $\quad O(n^2 \log n)$

O_y: 与 y 有关

4) $n\log_3 n + n\log_2 n$ $\quad O(n\log n)$

5) $n^{100} + 2^n$ $\quad O(2^n)$

Prop 1.2 (cont.) 3) If $f_1 = O(g_1)$ and $f_2 = O(g_2)$, then $f_1 + f_2 = O(\max(g_1, g_2))$

4) If $f = O(g)$ and k is a constant, $k \cdot f = O(g)$.

4) $(\exists c)(\exists N)(\forall x \geq N)(|f(x)| \leq c|g(x)|)$

$(\exists c' = kc)(\exists N)(\forall x \geq N)(|kf| \leq ck|g(x)|)$

Def 1.3 Let $f: \mathbb{R}^2 \to \mathbb{R}$ and $g: \mathbb{R} \to \mathbb{R}$. Write $f(x,y) = O_y(g(x))$ if

$\underbrace{(\forall y)(\exists c > 0)}_{\exists c = c(y) > 0}(\exists N)(\forall x \geq N)(|f(x,y)| \leq c \cdot |g(x)|)$

$O(1)$ universal constant $\qquad O_k(1)$: const dep. on k

$\frac{1}{1^2} + \frac{1}{2^2} + \cdots + \frac{1}{n^2} = O(1)$

Let $\delta > 0$, $\frac{1}{1^{1+\delta}} + \frac{1}{2^{1+\delta}} + \cdots + \frac{1}{n^{1+\delta}} = O_\delta(1)$

$< 1 + \sum_{i=2}^{n} \frac{1}{i^{1+\delta}} \leq 1 + \sum_{i=2}^{n} \int_{i-1}^{i} \frac{1}{x^{1+\delta}}\,dx = 1 + \int_1^n \frac{1}{x^{1+\delta}}\,dx = 1 + \left.\frac{x^{-\delta}}{-\delta}\right|_1^n \leq 1 + \frac{1}{\delta} = O_\delta(1)$

$1 + 2 + 3 + \cdots + n = \frac{1}{2}n(n+1) \overset{\in}{=} O(n^2)$

$\overset{\in}{=} \frac{1}{2}n^2 + O(n)$

$1^2 + 2^2 + \cdots + n^2 = \frac{1}{6}n(n+1)(2n+1) = O(n^3)$

$= \frac{1}{3}n^3 + O(n^2)$

$1^k + 2^k + \cdots + n^k = \frac{1}{k+1}n^{k+1} + O_k(n^k)$

$O(1)$ constant

$O(\log n)$ logarithmic

$O(\log^c n)$ polylog

$O(n)$ linear

$O(n \log^c n)$ quasilinear

$O(n^2)$ quadratic

$O(n^c)$ or $n^{O(1)}$ polynomial

$2^{(\log n)^{O(1)}}$ quasipolynomial

$O(c^n)$ or $2^{O(n)}$ exponential

Def 1.4 (Ω) Let $f, g: \mathbb{R} \to \mathbb{R}$. Write $f(x) = \Omega(g(x))$ if

$(\exists c > 0)(\exists N)(\forall x \geq N)(|f(x)| \geq c \cdot |g(x)|)$

O: $\leq$

*Chernoff bound. $X = x_1 + \cdots + x_n$, $x_i \in \{-1, 1\}$

$\Pr[X \geq k\sqrt{n}] \leq 2^{-\Omega(k^2)}$

Def 1.5 (Θ) Let $f, g: \mathbb{R} \to \mathbb{R}$, Write $f = \Theta(g)$ if

$(\exists c_1, c_2 > 0)(\exists N)(\forall x \geq N)(c_1 \cdot |g(x)| \leq |f(x)| \leq c_2 |g(x)|)$

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

	worst case	
quick sort	$O(n^2)$	
bubble sort	$O(n^2)$	$\Theta(n^2)$
Optimized bubble sort	$O(n^2)$	

Def 1.6 (Little o notation) Let $f, g: \mathbb{R} \to \mathbb{R}$. Write $f(x) = o(g(x))$

$(\forall \varepsilon > 0)(\exists N)(\forall x \geq N)(|f(x)| \leq \varepsilon |g(x)|)$

$2n = o(n^2)$, $2n = o(n^3)$

$o(1) \to 0$? 可以无极限

$\frac{1}{n}, \frac{1}{\log n}, \frac{1}{n^2}, \frac{1}{\log\log n} = o(1)$ $\quad (-1)^n \cdot \frac{1}{n} = o(1)$

$\frac{1}{2^{100}} \overset{?}{=} o(1)$ X $\quad \varepsilon = \frac{1}{2^{100}}$ $\quad \frac{1}{2^{100}} = O(1)$

Def 1.7 (ω) Let $f, g: \mathbb{R} \to \mathbb{R}$, Write $f(x) = \omega(g(x))$ if

$(\forall c > 0)(\exists N)(\forall x \geq N)(|f(x)| > c \cdot |g(x)|)$

想表述 $\frac{|f(x)|}{|g(x)|} \to 1$, 但避免 $g(x) = 0$

Def 1.8 (θ) Write $f(x) = \theta(g(x))$ if $\quad f \sim g$

$(\forall \varepsilon > 0)(\exists N)(\forall x \geq N)(||f(x)| - |g(x)|| \leq \varepsilon \cdot |g(x)|)$

Analogy

$f = O(g)$	$\leq$	$f = o(g)$	$<$
$f = \Omega(g)$	$\geq$	$f = \omega(g)$	$>$
$f = \Theta(g)$	$\approx$	$f = \theta(g)$	$=$

1) $f(n) = 100n\log n$, $g(n) = n^2$

$f(n) = O(g(n))$. $f(n) = o(g(n))$

$g(n) = \Omega(g(n))$, $g(n) = \omega(f(n))$

2) $f(n) = 6n^2 + 100n$, $g(n) = n^2$

$f(n) = O(g(n))$, $g(n) = \Omega(f(n))$, $f(n) = \Theta(g(n))$

3) $f(n) = n^{100}$, $g(n) = 2^n$

$f(n) = o(g)$, $f = O(g)$ $\quad g = \omega(f)$, $g = \Omega(f)$

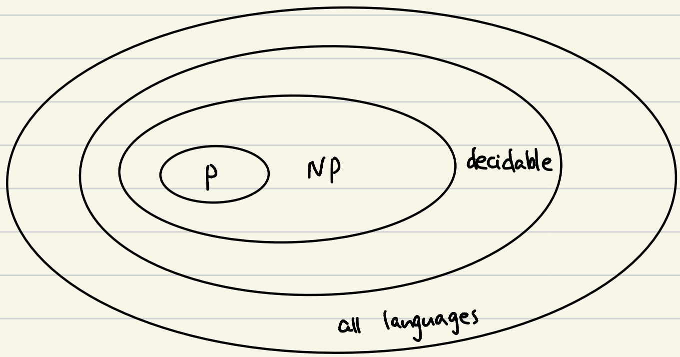
4) $f(n) = n^{O(1)}$ or poly(n)

$(\exists c > 0)(\exists N)(\forall n \geq N)(|f(n)| \leq n^c)$

Def 1.9 ($\tilde{O}$) Let $f, g : \mathbb{R} \to \mathbb{R}$. Write $f(x) = \tilde{O}(g(x))$ if

$(\exists c > 0)(\exists N)(\forall x \geq N)(|f(x)| \leq (\log n)^c |g(x)|)$

$\tilde{O}$ hides polygarithmic term $O((\log n)^c n)$ $\tilde{O}(n)$ quasilinear



1.2 Alphabets and languages

Def 1.10 An <u>alphabet</u> is a set of symbols. Σ

Roman alphabet $\{a, b, \cdots, z\}$

Binary alphabet $\{0,1\}$.

Def 1.11 A <u>string</u> over an alphabet is a finite sequence of symbols from the alphabet.

empty string ε

Def 1.12 The set of all string over the alphabet Σ is denoted by Σ^*

Denote by Σ^n the set of strings of length n.

$\Sigma^* = \varepsilon \cup \Sigma \cup \Sigma^2 \cup \cdots$

$\{0,1\}^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, 000, \cdots\}$

The length of a string is its length as a sequence

Def 1.13 (concatenation) Two strings over the same alphabet can be combined to form a third by <u>concatenation</u>.

The concatenation of strings x and y is denoted by xy.

Def 1.14 (language) A <u>language</u> is a set of strings over an alphabet.

$L \subseteq \Sigma^*$

$\Sigma = \{0,1\}$ $\quad \phi, \Sigma^*$

Even $= \{0, 10, 100, 110, \cdots\}$

Prime $= \{2, 3, \cancel{5, 7}, 11, \cdots\} = \{10, 11, 101, 111, 1011\}$

Prime' $= \{11, 111, 11111, \cdots\}$ X 编码不合理

Palindrome $= \{0, 00, 11, 101, 000, 111, \cdots\}$

Def 1.15 (complement) Let $L \subseteq \Sigma^*$. The complement of L denoted by $\overline{L}$

$\Sigma^* \setminus L$. So, $\overline{\overline{L}} = L$.

union $\cup$, intersection $\cap$, difference $\setminus$.

If $L_1, L_2 \subseteq \Sigma^*$, the concatenation of L_1 and L_2 is

$L_1L_2 = \{w \in \Sigma^* : w = xy \text{ for some } x \in L_1, y \in L_2\}$.

Def 1.16 (star) Let $L \subseteq \Sigma^*$. The Kleene star of L, denoted by L^*, is the set of strings obtained by concatenating zero or more strings from L. That is $L^* = \{w \in \Sigma^* : w = w_1 \cdots w_k \text{ for } k \geq 0 \text{ and } w_1, \cdots, w_k \in L\}$

Write L^+ for LL^*. // L^+: 不包含空 string

$L = \{1\}$. $L^* = \{\varepsilon, 1, 11, 111, \cdots\}$

9.17

ToC

1. Intro — Big O notation, language
2. Finite automaton 有限自动机
3. Turing Machine

} What is computation?

4. Computability — What can be computed?
5. Complexity — What can be computed efficiently?

1.3 Encoding of problems

Problem encoding is the process of representing inputs, outputs using strings over a fixed alphabet, typically $\{0,1\}$.

Example

1) (Integer multiplication) Given two nonnegative integers x, y.

   compute $x \cdot y$ $\qquad f: \{0,1\}^* \to \{0,1\}^* \cup \{\text{undefined}\}$

   input 5, 10

   output 50

2) (Primality testing) Given $n \in \mathbb{N}$, decide if n is a prime.

   Input: 101 (5) Input: 100 (4)

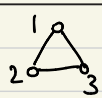
   Output: Yes Output: NO

   Binary string, most significant bit first

   $L = \{ n \in \{0,1\}^* : n = n_1 n_2 \cdots n_t \in \{0,1\}^*, n_i \in \{0,1\}$

   $n_1 \cdot 2^{t-1} + n_2 \cdot 2^{t-2} + \cdots + n_t \cdot 2^0$ is a prime$\}$

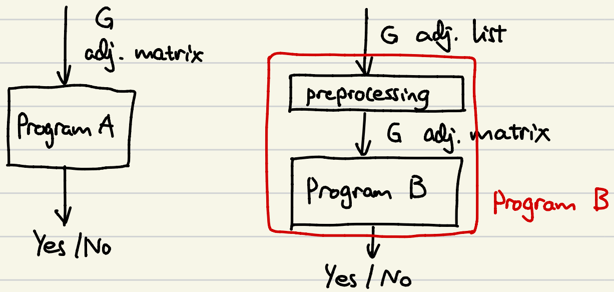
3) (Hamilitonian Cycle) Given a graph G, decide if G has a Hamiltonian cycle.

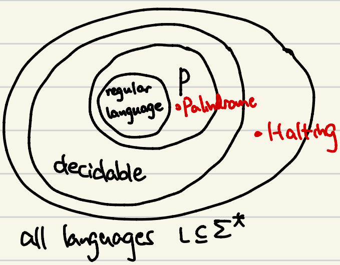


adj. matrix	adj. list
0 1 1	1␣2
1 0 1	1␣3
1 1 0	2␣3

无向, 对称

With preprocessing, we can switch between encodings.





Decision Problem $\quad L \subseteq \{0,1\}^*$

Computation problem $\quad f: \{0,1\}^* \to \{0,1\}^* \cup \{\perp\}$

Hereafter, when we say a decision, e.g. SAT, Halting Problem, we mean the language of that problem by means of a reasonable encoding.

regular expression $\longrightarrow$ Finite automaton.