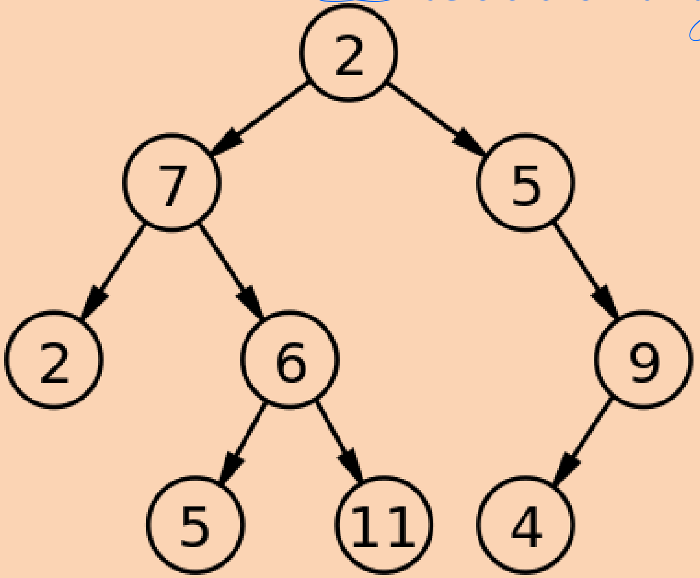
ساختمان های داده

تحلیل کارایی الگوریتم ها- کشف نمادهای مجانبی

Performance Analysis of Algorithms-Discovering Asymptotic Notations



مدرس:

سیدکمال الدین غیاثی شیرازی

نتیجه مبحث قبل

- در مقایسه الگوریتم ها سریع تر بودن به اندازه یک ضریب ثابت را نادیده می گیریم.
- زمان اجرا را برای داده های ورودی بزرگ در نظر می گیریم.
- برای داده های ورودی با اندازه یکسان دو تحلیل داریم:
  - **تحلیل بدترین حالت**
  - **تحلیل زمان متوسط**

تحلیل مجانبی

- از آنجا که زمان اجرا برای داده های کوچک اهمیتی ندارد، فرض می کنیم n_0 ای وجود

$\exists n_0 \quad \forall n > n_0$

دارد که زمان اجرا برای داده های بزرگ تر از n_0 برایمان اهمیت دارد.

- از آنجا که سریع‌تر بودن به اندازه یک ضریب ثابت اهمیتی ندارد، پس از $g(n) > f(n)$ نمی‌توان نتیجه گرفت که الگوریتمی با تعداد گام $f(n)$ از الگوریتمی با تعداد گام $g(n)$ سریع‌تر است.

مثال: الگوریتم مرتب‌سازی حبابی

```
n = length(A)
for j = n downto 1 do
    for i = 1 to j-1 do
     if A[i] > A[i+1] then
       swap(A[i], A[i+1])
```

1   4

1   3

1   4

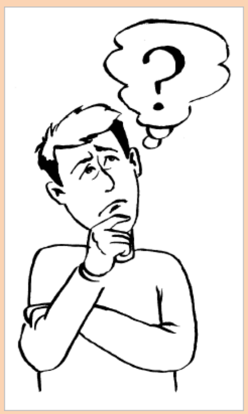
تحلیل بدترین حالت: داده برعکس مرتب باشه

$$4 + \sum_{j=1}^{n} \sum_{i=1}^{j-1} 7 = 4 + \frac{7(n)(n-1)}{2}$$

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

چه زمانی زمان اجرای دو الگوریتم مساوی است؟

- فرض کنیم الگوریتم A زمان اجرای $f(n)$ و الگوریتم B زمان اجرای $g(n)$ را دارد.
- می گوییم زمان اجرای مجانبی الگوریتم A با الگوریتم B برابر است هرگاه



برای n های بزرگ با یک ضریب ثابت $f(n)$ بزرگتر از $g(n)$ باشه

$$c_1 g(n) < f(n)$$

و با ضریب ثابت دیگری، $g(n)$ بزرگتر از $f(n)$ باشه

$$c_2 f(n) < g(n)$$

$$c_1 f(n) \leqslant g(n) \leqslant c_2 f(n)$$

$$\exists c_1 > 0, c_2 > 0 \quad \exists n_0 \in N \quad \forall n > n_0$$

چه زمانی زمان اجرای دو الگوریتم مساوی است؟

- فرض کنیم الگوریتم A زمان اجرای $f(n)$ و الگوریتم B زمان اجرای $g(n)$ را دارد.
- می گوییم زمان اجرای مجانبی الگوریتم A با الگوریتم B برابر است هرگاه با انتخاب ضرایب ثابت مناسب هم $\mathrm{f(n)} \leq \mathrm{c_1 g(n)}$ و هم $g(n) \leq c_2 f(n)$ باشد.
- به زبان ریاضی:

$c_1' = \frac{1}{c_1}$

$$\exists c_1 > 0\ \exists c_2 > 0\ \exists n_0 \in \mathbb{N}\ \forall n \geq n_0\ \overbrace{f(n) \leq c_1 g(n),\ \ g(n) \leq c_2 f(n)}^{c_1' f(n) \leq g(n) \leq c_2 f(n)}$$

- در این صورت می نویسیم: $g(n) \in \Theta(f(n))$

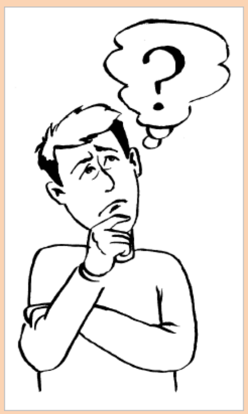
$f(n) \in \Theta(g(n))$

چه زمانی زمان اجرای الگوریتمی از الگوریتم دیگر اکیداً کمتر است؟

- فرض کنیم الگوریتم A زمان اجرای $f(n)$ و الگوریتم B زمان اجرای $g(n)$ را دارد.

$\cancel{f(n) < g(n)}$

- می گوییم زمان اجرای الگوریتم A از الگوریتم B کمتر است هرگاه:



هر قدر هم که ثابت $c>0$ را کوچک انتخاب کنیم

باز هم برای n های به اندازه کافی بزرگ

$$f(n) < c\,g(n)$$

به عبارت دیگر کمتر بودن $f(n)$ با یک ضریب ثابت جبران نمی‌شود

$$\forall c>0 \quad \exists n_0 \in \mathbb{N} \quad \forall n > n_0 \quad f(n) < c\,g(n)$$

$$\forall c>0 \quad \exists n_0 \in \mathbb{N} \quad \forall n > n_0 \quad c\,f(n) < g(n)$$

چه زمانی زمان اجرای الگوریتمی از الگوریتم دیگر اکیداً کمتر است؟

- فرض کنیم الگوریتم A زمان اجرای $f(n)$ و الگوریتم B زمان اجرای $g(n)$ را دارد.
- می گوییم زمان اجرای الگوریتم A از الگوریتم B کمتر است هرگاه:
- هرقدر هم که ثابت c بزرگ باشد، باز هم برای n های به اندازه کافی بزرگ داشته باشیم:

$$cf(n) < g(n)$$

- به زبان ریاضی:

$$\forall c > 0\ \exists n_0 \in \mathbb{N}\ \forall n \geq n_0\ cf(n) \leq g(n)$$

- در این صورت می نویسیم: $f(n) \in o\big(g(n)\big)$

آزمونك

هر کسی دوستی دارد $\forall n\ \exists y$ y دوست n است

کسی هست که دوست همه است $\exists y\ \forall n$ y دوست n است

- کدام یک از تعاریف زیر برای $f(n) \in o\big(g(n)\big)$ صحیح است؟

هر قدر هم c بزرگ باشه

$\forall c > 0\ \exists n_0 \in \mathbb{N}\ \forall n > n_0\ cf(n) < g(n)$ ✓

$\forall c > 0\ \exists n_0 \in \mathbb{N}\ \forall n > n_0\ cf(n) \leq g(n)$ ✓

$\forall c > 0\ \exists n_0 \in \mathbb{N}\ \forall n \geq n_0\ cf(n) < g(n)$ ✓

$\forall c > 0\ \exists n_0 \in \mathbb{N}\ \forall n \geq n_0\ cf(n) \leq g(n)$ ✓

هر قدر هم c کوچک باشه

$\forall c > 0\ \exists n_0 \in \mathbb{N}\ \forall n > n_0\ f(n) < cg(n)$ ✓

$\forall c > 0\ \exists n_0 \in \mathbb{N}\ \forall n > n_0\ f(n) \leq cg(n)$ ✓

$\forall c > 0\ \exists n_0 \in \mathbb{N}\ \forall n \geq n_0\ f(n) < cg(n)$ ✓

$\forall c > 0\ \exists n_0 \in \mathbb{N}\ \forall n \geq n_0\ f(n) \leq cg(n)$ ✓

$\forall c \geq 0\ \exists n_0 \in \mathbb{N}\ \forall n > n_0\ cf(n) < g(n)$ ✗

$\exists n_0 \in \mathbb{N}\ \forall c \geq 0\ \forall n > n_0\ cf(n) < g(n)$ ✗

اگر c به اندازه کافی بزرگ انتخاب شود ← $cf(n_0) > g(n_0)$

چه زمانی زمان اجرای الگوریتمی از الگوریتم دیگر اکیداً بیشتر است؟

- فرض کنیم الگوریتم A زمان اجرای f(n) و الگوریتم B زمان اجرای g(n) را دارد.

اکیداً

- می گوییم زمان اجرای الگوریتم A از الگوریتم B بیشتر است هرگاه زمان اجرای الگوریتم B

اکیداً

از الگوریتم A کمتر باشد.

$f(n) \in \omega(g(n)) \Longleftrightarrow g(n) \in o(f(n))$

اکیداً بیشتر

- به زبان ریاضی:

$$\forall c > 0\ \exists n_0 \in \mathbb{N}\ \forall n \geq n_0\ f(n) \geq cg(n)$$

- در این صورت می نویسیم: $f(n) \in \omega(g(n))$

آزمونک

اکیدآ بیشتر

- کدام یک از تعاریف زیر برای $f(n) \in \omega\big(g(n)\big)$ صحیح است؟

$$\forall c > 0\ \exists n_0 \in \mathbb{N}\ \forall n > n_0\ cg(n) < f(n)$$

$$\forall c > 0\ \exists n_0 \in \mathbb{N}\ \forall n > n_0\ cg(n) \leq f(n)$$

$$\forall c > 0\ \exists n_0 \in \mathbb{N}\ \forall n \geq n_0\ cg(n) < f(n)$$

$$\forall c > 0\ \exists n_0 \in \mathbb{N}\ \forall n \geq n_0\ cg(n) \leq f(n)$$

$$\forall c > 0\ \exists n_0 \in \mathbb{N}\ \forall n > n_0\ g(n) < cf(n)$$

$$\forall c > 0\ \exists n_0 \in \mathbb{N}\ \forall n > n_0\ g(n) \leq cf(n)$$

$$\forall c > 0\ \exists n_0 \in \mathbb{N}\ \forall n \geq n_0\ g(n) < cf(n)$$

$$\forall c > 0\ \exists n_0 \in \mathbb{N}\ \forall n \geq n_0\ g(n) \leq cf(n)$$

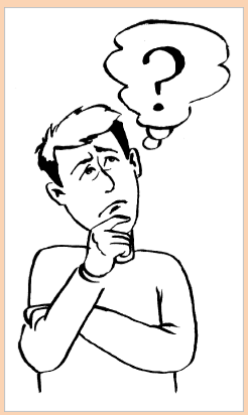
$$\forall c \cancel{\geq} 0\ \exists n_0 \in \mathbb{N}\ \forall n > n_0\ cg(n) < f(n)$$

$$\exists n_0 \in \mathbb{N}\ \forall c \geq 0\ \forall n > n_0\ cg(n) < f(n)$$

نماد O به مفهوم کوچک‌تر مساوی

- فرض کنیم الگوریتم A زمان اجرای $f(n)$ و الگوریتم B زمان اجرای $g(n)$ را دارد.

- می گوییم زمان اجرای الگوریتم A کوچک‌تر یا مساوی الگوریتم B است هرگاه:



کوچکتر مساوی $\neq$ کوچکتر یا مساوی

$$f(n)\in O(g(n)) \not\Leftrightarrow f(n)\in o(g(n)) \vee f(n)\in\Theta(g(n))$$

$$f(n)\in o(g(n)) \Rightarrow f(n)\in O(g(n))$$

$$f(n)\in \Theta(g(n)) \Rightarrow f(n)\in O(g(n))$$

$f(n)\in O(g(n))$ هرگاه $n_0\in\mathbb{N}$ وجود داشته باشد و $c>0$ وجود داشته باشد به طوری که

$$\forall n > n_0 \quad f(n)\leq c\,g(n)$$

نماد O به مفهوم کوچک تر مساوی

- فرض کنیم الگوریتم A زمان اجرای $f(n)$ و الگوریتم B زمان اجرای $g(n)$ را دارد.
- می گوییم زمان اجرای الگوریتم A کوچک‌تر یا مساوی الگوریتم B است هرگاه:

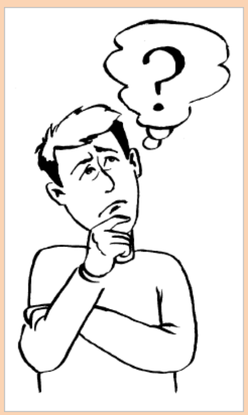
$$\exists c > 0\ \exists n_0 \in \mathbb{N}\ \forall n \geq n_0\ \ f(n) \leq cg(n)$$

و می نویسیم: $f(n) \in O(g(n))$

$$\exists c>0\ \exists n_0 \in \mathbb{N}\ \forall n \geq n_0\ \ cf(n) \leq g(n)$$

نماد Ω به مفهوم بزرگ‌تر مساوی

- فرض کنیم الگوریتم A زمان اجرای $f(n)$ و الگوریتم B زمان اجرای $g(n)$ را دارد.
- می گوییم زمان اجرای الگوریتم A بزرگ‌تر یا مساوی الگوریتم B است هرگاه:



$$f(n) \in \Omega(g(n)) \iff g(n) \in O(f(n))$$

$$\exists c > 0 \quad \exists n_0 \in \mathbb{N} \quad \forall n > n_0 \qquad cf(n) \geq g(n)$$

$$cf(n) > g(n)$$

$$f(n) > c\,g(n)$$

$$f(n) \geq c\,g(n)$$

نماد Ω به مفهوم بزرگ‌تر مساوی

- فرض کنیم الگوریتم A زمان اجرای $f(n)$ و الگوریتم B زمان اجرای $g(n)$ را دارد.
- می گوییم زمان اجرای الگوریتم A بزرگ‌تر یا مساوی الگوریتم B است هرگاه:

$$\exists c > 0 \ \exists n_0 \in \mathbb{N} \ \forall n \geq n_0 \ \ f(n) \geq cg(n)$$

و می نویسیم: $f(n) \in \Omega(g(n))$

شکلی از کتاب CLRS

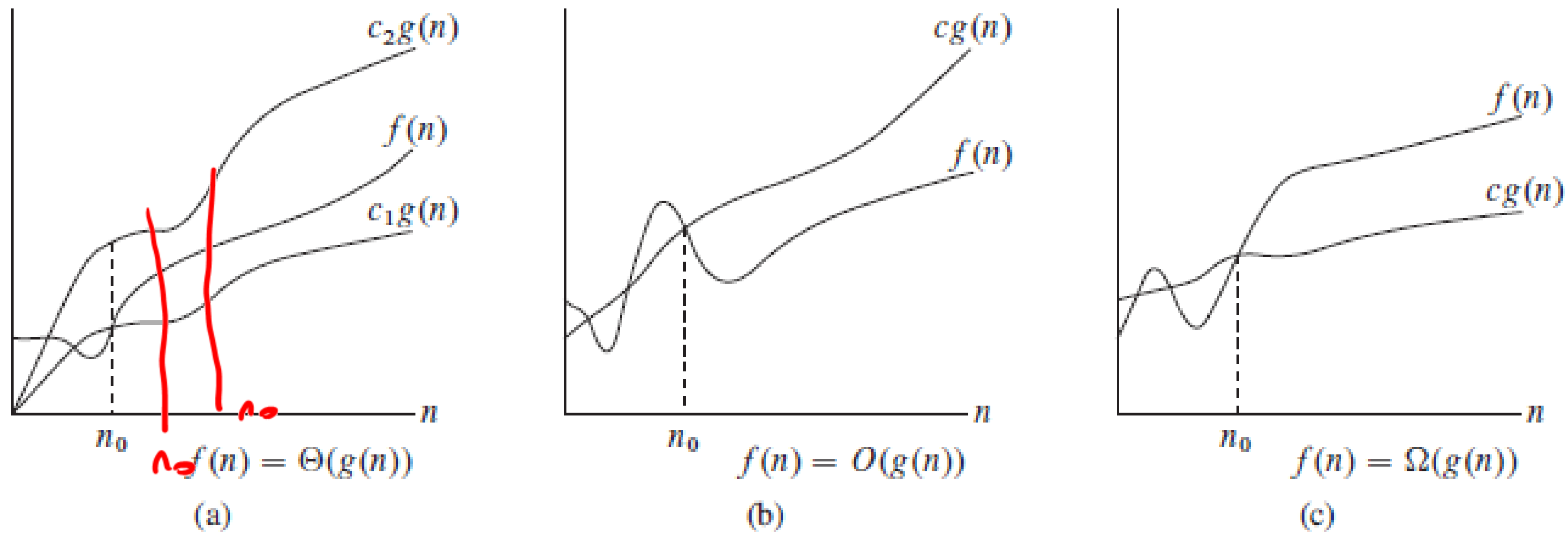


Figure 3.1 Graphic examples of the Θ, O, and Ω notations. In each part, the value of n_0 shown is the minimum possible value; any greater value would also work. (a) Θ notation bounds a function to within constant factors. We write $f(n) = \Theta(g(n))$ if there exist positive constants n_0, c_1, and c_2 such that at and to the right of n_0, the value of $f(n)$ always lies between $c_1g(n)$ and $c_2g(n)$ inclusive. (b) O notation gives an upper bound for a function to within a constant factor. We write $f(n) = O(g(n))$ if there are positive constants n_0 and c such that at and to the right of n_0, the value of $f(n)$ always lies on or below $cg(n)$. (c) Ω notation gives a lower bound for a function to within a constant factor. We write $f(n) = \Omega(g(n))$ if there are positive constants n_0 and c such that at and to the right of n_0, the value of $f(n)$ always lies on or above $cg(n)$.

$cf(n)$
$g(n)$

بیان مجموعه ای

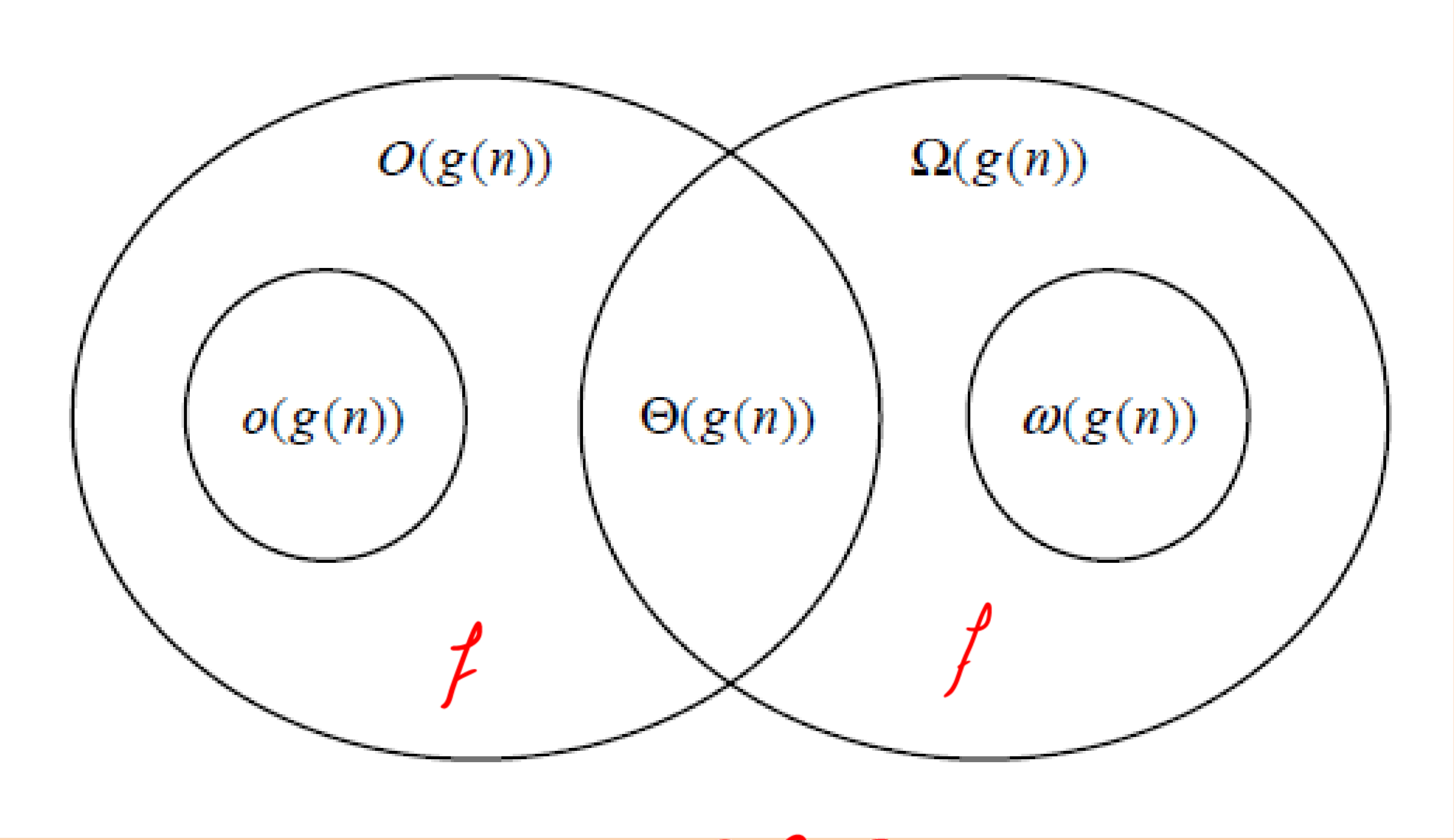
- برای هر تابع $f(n)$ می‌توان $o(f(n))$، $O(f(n))$، $\Theta(f(n))$، $\Omega(f(n))$ و $\omega(f(n))$ را به صورت مجموعه‌هایی تعریف کرد که اعضای آنها توابعی هستند که رشد آنها به ترتیب کوچک تر، کوچک تر مساوی، مساوی، بزرگ تر مساوی و بزرگ تر از تابع $f(n)$ است.

$\leq$	$<$	$=$	$>$	$\geq$
O	o	Θ	ω	Ω

$f \in O(g)$

هم ارزی

رابطه مجموعه های تحلیل مجانبی



$\Theta = O \cap \Omega$

تعیین مرتبه زمانی اجرای الگوریتم

- با تحلیل مجانبی ، علاوه بر مقایسه الگوریتم ها ، می‌توانیم زمان اجرای الگوریتم را بر حسب توابع ریاضی نیز بیان کنیم.
- **مثال:** وقتی می گوییم مرتبه زمانی اجرای الگوریتم A از $O(n^3)$ است، یعنی: اگر زمان اجرای الگوریتم A برابر تابع $f(n)$ باشد، آنگاه: $f(n) \in O(n^3)$

Θ O

نمادهای مهم تر

- از آنجا که معمولا ما بیشتر به حد بالای زمان اجرای الگوریتم ها علاقه مند هستیم ، نماد O بیشترین کاربرد را دارد.

- فرض کنیم زمان اجرای الگوریتمی برابر n است. در این صورت این الگوریتم علاوه بر اینکه از مرتبه $O(n)$ است ، از مرتبه های $O(n\,logn)$ و $\mathrm{O}(\mathrm{n}^{10})$ نیز می باشد.

  - نماد Θ می‌گوید که حد بالایی که برای زمان اجرای الگوریتم پیدا کرده‌ایم ، حد پایین هم هست.

  - زمان اجرای الگوریتم از مرتبه $\Theta(n)$ است ولی از مرتبه‌های $\Theta(nlogn)$ و $\Theta(n^{10})$ نیست.

آزمونك

$\Theta(f(n)) = \{g(n) | \exists c_1, c_2 > 0 \; \exists n_0 \in \mathbb{N} \; \forall n > n_0 \quad c_1 f(n) \leq g(n) \leq c_2 f(n) \}$

$o(f(n)) = \{g(n) | \forall c > 0 \; \exists n_0 \in \mathbb{N} \; \forall n > n_0 \quad c\, g(n) \leq f(n) \}$

$O(f(n)) = \{g(n) | \exists c > 0 \; \exists n_0 \in \mathbb{N} \; \forall n > n_0 \quad g(n) \leq c f(n) \}$

$\omega(f(n)) = \{g(n) | \forall c > 0 \; \exists n_0 \in \mathbb{N} \; \forall n > n_0 \quad c f(n) \leq g(n) \}$

$\Omega(f(n)) = \{g(n) | \exists c > 0 \; \exists n_0 \in \mathbb{N} \; \forall n > n_0 \}$

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

یادآوری تعاریف

$$\Theta\big(f(n)\big) = \{g(n) | \exists c_1, c_2 > 0\ \exists n_0 \in \mathbb{N}:\ \forall n \geq n_0\ c_1 g(n) \leq f(n) \leq c_2 g(n)\}$$

$$o\big(f(n)\big) = \{g(n) | \forall c > 0\ \exists n_0 \in \mathbb{N}:\ \forall n \geq n_0\ cg(n) \leq f(n)\}$$

$$O\big(f(n)\big) = \{g(n) | \exists c > 0\ \exists n_0 \in \mathbb{N}: \forall n \geq n_0\ cg(n) \leq f(n)\}$$

$$\omega\big(f(n)\big) = \{g(n) | \forall c > 0\ \exists n_0 \in \mathbb{N}:\ \forall n \geq n_0\ g(n) \geq cf(n)\}$$

$$\Omega\big(f(n)\big) = \{g(n) | \exists c > 0\ \exists n_0 \in \mathbb{N}:\ \forall n \geq n_0\ g(n) \geq cf(n)\}$$

تمرین

- با استفاده از تعریف ثابت کنید:

- $n + 5 \in \Theta(n)$
- $n \log n \in O(n^2)$

$\exists \underbrace{c_1}_{1}, \underbrace{c_2}_{2} > 0 \quad \exists \underbrace{n_0}_{5} \in \mathbb{N} \quad \forall n \geq n_0 \quad c_1 n \leq n+5 \leq c_2 n$

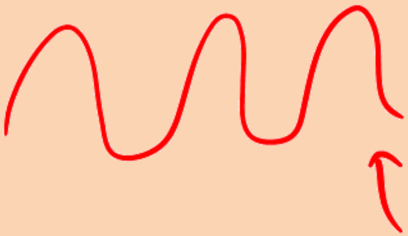
$\exists \underbrace{c}_{1} > 0 \quad \exists \underbrace{n_0}_{1} \in \mathbb{N} \quad \forall n > n_0 \quad c\, n \log n \leq n^2$

$c \log n \leq n$

قضیه

$+\infty$

- فرض کنید حد زیر وجود دارد و متناهی یا بی نهایت است.



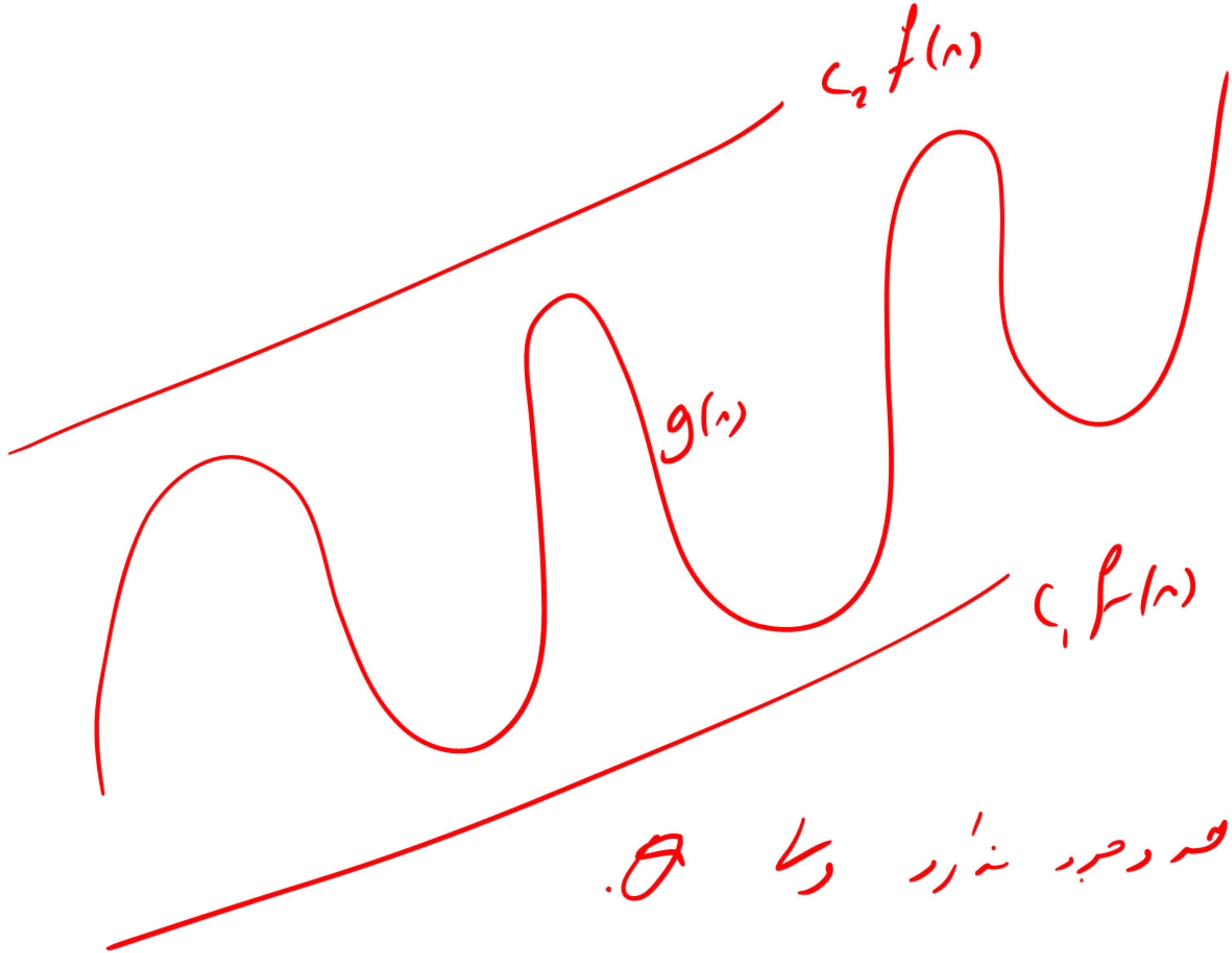
نوسان ⇐ حد وجود ندارد

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

- $f \in \Theta(g)$ اگر و فقط اگر $0 < L < \infty$
- $f \in o(g)$ اگر و فقط اگر $L = 0$
- $f \in \omega(g)$ اگر و فقط اگر $L = \infty$

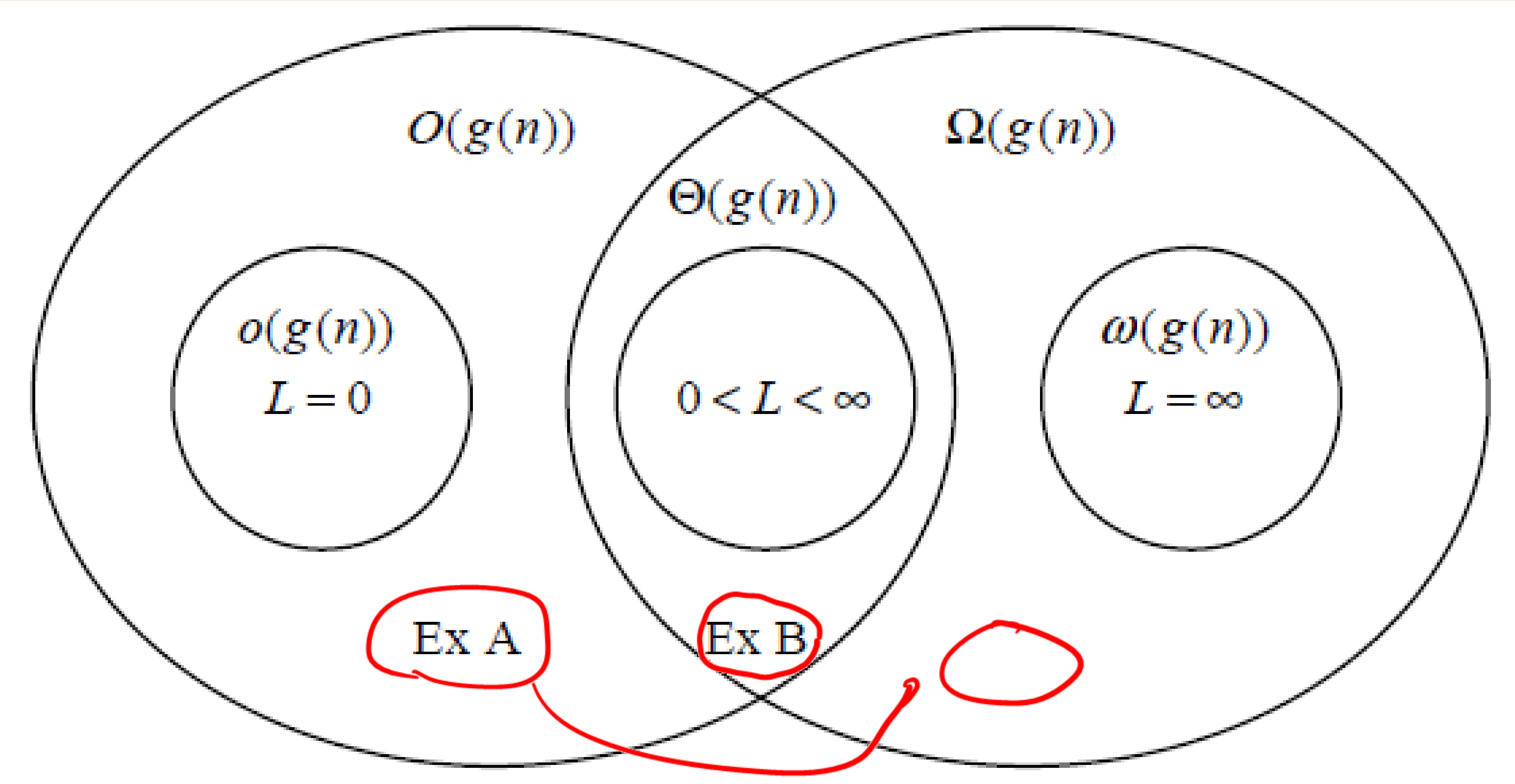
اگر $f \in o(g)$ آنگاه $\lim\limits_{n\to\infty} \frac{f(n)}{g(n)} = 0$

ممکن است $f \in \theta(g)$ و یا حد وجود نداشته باشد.



حد وجود ندارد ولی Θ.

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



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

با فرض $f(n)$ و $g(n)$ برای nهای بزرگ مثبت باشند

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0 \iff \forall \varepsilon > 0 \ \exists M \in \mathbb{N} \ \forall n > M \ \left| \frac{f(n)}{g(n)} - 0 \right| < \varepsilon$$

برای تعیین مثبت بودن g و f

$$\iff \forall \underbrace{\varepsilon}_{c} > 0 \ \exists \underbrace{M}_{n_0} \in \mathbb{N} \ \forall n > M \ f(n) < \varepsilon g(n)$$

$$\iff \forall c > 0 \ \exists n_0 \in \mathbb{N} \ \forall n > n_0 \ f(n) < c\, g(n)$$

$$\iff f \in o(g)$$

مثالی از $f \in O(g)$ که $f \notin o(g)$ و $f \notin \Theta(g)$

$$n \in o(n^2)$$
$$n^2 \in \Theta(n^2)$$

$$f(n) = \begin{cases} n & \text{زوج } n \\ n^2 & \text{فرد } n \end{cases}$$

$$f(n) \in O(n^2)$$
$$f(n) \notin o(n^2)$$
$$f(n) \notin \Theta(n^2)$$