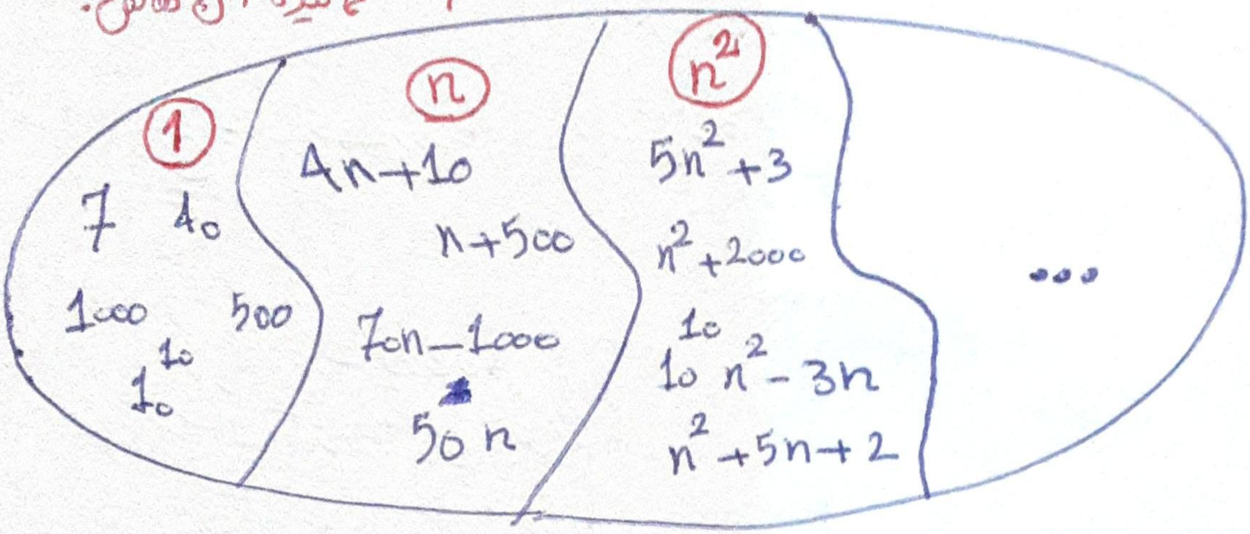
کلاس بندی توابع: تعداد توابع زیادی وجود دارند و از اینرو برای مقایسه آنها ترجیح می‌دهیم توابع را بدون از دست دادن حقیقت ماهیتشان، کلاس بندی کنیم.

معرفی یک عضو از کلاس به عنوان نماینده آن کلاس.



نکته: فرض کنید $f(n)$ بیانگر تابع پیچیدگی زمانی الگوریتم ما باشد، همواره سعی در آن است که در دسته بندی و مقایسه $f(n)$ با سایر توابع به صورت زیر عمل کنیم:

* $f(n)=O(g(n))$ ← $g(n)$ کوچکترین حد بالا برای $f(n)$ باشد.

مثال: $f(n)=5n^2+3$

$f(n)=O(n^2)$ ✓

درحالیکه طبق تعریف موارد زیر نیز درست.

$f(n)=O(n^3)$, $f(n)=O(n^4)$, $f(n)=O(n\log n)$, $f(n)=O(2^n)$, ...

* $f(n)=\Omega(g(n))$ ← $g(n)$ بزرگترین حد پایین برای $f(n)$ باشد.

مثال: $f(n)=5n^3+4n$

$f(n)=\Omega(n^3)$ ✓

در صورتیکه طبق تعریف موارد روبرو هم درست است: $f(n)=\Omega(n^2)$, $f(n)=\Omega(n)$, ...

سؤال: آیا شبیه به اعداد حقیقی که هر زوج مقدار a و b در آن در یک وضعیت

$a>b$

$a<b$

$a=b$

است، بین دو تابع $f(n)$ و $g(n)$ هم یک وضعیت

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

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

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

را همواره می‌توان بررسی کرد؟

پاسخ: جواب منفی است. مثال:

$f(n)=n$

$g(n)=n^{1+\sin n}$ → توان این عبارت بین 0 و 2 نوسان می‌کند و بنابراین نمی‌توانیم طبق تعاریف بیان شده برای n ها c, n_0 ای پیدا کنیم که شرایط را برآورده کند.

---

* رابطه بین مفاهیم پیچیدگی زمانی در

بهترین حالت

بدترین حالت

حالت متوسط

و نمادهای

Ω

O

Θ

شاید در نگاه اول این به ذهن برسد که بهترین حالت معادل Ω

بدترین حالت معادل O

حالت متوسط معادل Θ

اما

با رجوع به تعاریف در خواهید یافت که نمادها در حالت کلی برای مقایسه نرخ رشد توابع نسبت به یکدیگر است در حالیکه حالت های بهترین، بدترین و متوسط برای زمان اجرای یک الگوریتم متناسب با نمونه ورودی های آن می باشند.

مثال هایی برای نمادهای معرفی شده:

مثال ۱: دو تابع زیر را در نظر بگیرید:

$$f(n)=\begin{cases} 2^n & n\leq 10^{10} \\ 100n & n>10^{10}\end{cases} \quad , \quad g(n)=2n^2+n$$

روابط زیر را بررسی کنید:

الف) : $f(n) \overset{?}{=} O(g(n))$،

ب) : $g(n) \overset{?}{=} O(f(n))$

$\log n! = O(n\log n)$ and $\log n! = \Omega(n\log n)$

واضح است — از راهنمایی استفاده کنید

مثال ۲: نشان دهید که $\log n! = \Theta(n\log n)$.

راهنمایی: تقریب استرلینگ

$$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{\alpha_n}, \text{ where } \frac{1}{12n+1} < \alpha_n < \frac{1}{12n}$$

مثال ۳: فرض کنید $f(n)=O(g(n))$ باشد، آیا می‌توان رابطه $2^{f(n)} = O(2^{g(n)})$ را نتیجه گرفت؟ خیر ← مثال نقض:

$f(n)=2n$ $\qquad f(n)=O(g(n))$ ✓

$g(n)=n$ $\qquad 2^{f(n)} = O(2^{g(n)}) \rightarrow 2^{2n} = O(2^n)$ ✗

$(2^2)^n = 4^n \not\leq 2^n$ ✗

مثال ۴: فرض کنید $f(n) = O(g(n))$ باشد و $\log(g(n)) \geq 1$ و $f(n) \geq 1$ برای همه n های

بااندازه کافی بزرگ باشد. نشان دهید

$$\log(f(n)) = O(\log(g(n)))$$

از روی تعریف $f(n) = O(g(n))$:

$$\exists\, c, n_0 > 0 \quad s.t. \quad \forall n \geq n_0 \quad 0 \leq f(n) \leq c\,g(n)$$

طبق صورت سوال ← $1 \leq f(n) \leq c\,g(n)$

log بگیریم از دو طرف → $0 \leq \log(f(n)) \leq \log(c\,g(n))$

نکته $f(n) \geq 1$ ✓

تا اینجا را از تعریف $f(n) = O(g(n))$ داریم. ← (*) $0 \leq \log(f(n)) \leq \log c + \log(g(n))$

بررسی ادعای سوال: $\log(f(n)) = O(\log(g(n)))$

$$\exists\, d, n'_0 > 0 \quad s.t. \quad \forall n \geq n'_0 \quad 0 \leq \log(f(n)) \leq d\log(g(n))$$

براساس (*) ← $\log(f(n)) \leq \log c + \log(g(n)) \leq d\log(g(n))$

نکته: $\log(g(n)) \geq 1$ در صورت سوال

$$\begin{cases} d = \log(c) + 1 \\ n'_0 = n_0 \end{cases}$$