دانشکده علوم ریاضی و آمار

مدرس: دکتر مجتبی رفیعی نیمسال اول ۱۴۰۰-۱۴۰۱

ساختمان داده‌ها و الگوریتم‌ها

جلسه ۱۲

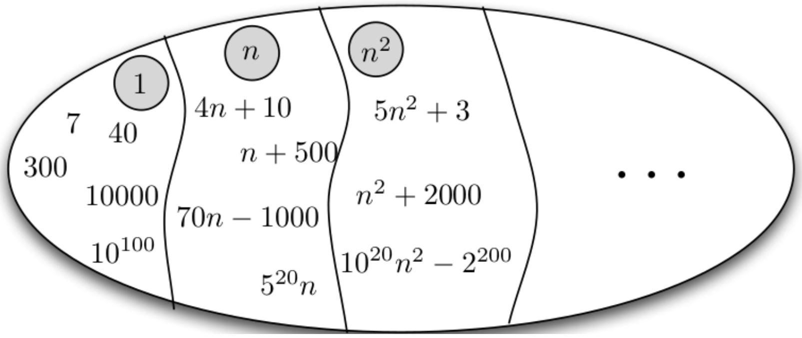
نگارنده: سپیده دوات ساز اصفهانی

۶ آبان ۱۴۰۰

فهرست مطالب

۱ کلاس‌بندی توابع

با توجه به اینکه توابع زیادی وجود دارد به **کلاس‌بندی** توابع می‌پردازیم. از هر کلاس یک عضو را به عنوان **نماینده کلاس** معرفی می‌کنیم و در نمادهای مجانبی از آن استفاده می‌کنیم. توجه داشته باشید که این کلاس‌بندی به هیچ عنوان سبب از بین رفتن خصیصه توابع نمی‌شود، در واقع نوعی ساده‌سازی است. مثلا به جای نوشتن $5n^2 + 3n + 2$ از نماینده‌اش یعنی $\mathcal{O}(n^2)$ استفاده می‌کنیم. شکل۱، یک نمای کلی از این کلاس‌بندی را نشان می‌دهد.



شکل ۱: کلاس‌بندی توابع

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

۱) برای رابطه $f(n) = O(g(n))$، تابع $g(n)$، **کوچکترین حد بالا** برای $f(n)$ باشد. به عنوان مثال اگر داشته باشیم $f(n) = 5n^2+3$، آنگاه $f(n) = \mathcal{O}(n^2)$ برای ما مطلوب است. با اینحال، طبق تعریف موارد زیر نیز صحیح است:

$$f(n) = \mathcal{O}(n^3),\ f(n) = \mathcal{O}(n^4),\ f(n) = \mathcal{O}(2^n).$$

۲) برای رابطه $f(n) = \Omega(g(n))$، تابع $g(n)$، **بزرگترین حد پایین** برای $f(n)$ باشد. به عنوان مثال اگر داشته باشیم $f(n) = 5n^2+4$، آنگاه $f(n) = \Omega(n^2)$ برای ما مطلوب است. با اینحال، طبق تعریف موارد زیر نیز صحیح است:

$$f(n) = \Omega(n \log n),\ f(n) = \Omega(n).$$

۲ مقایسه عملگرها در اعداد حقیقی و توابع

سوال: آیا شبیه به اعداد حقیقی که هر زوج a و b در آن در یکی از سه وضعیت زیر است:

- $a > b$
- $a = b$
- $a < b$

بین دو تابع $f(n)$ و $g(n)$ هم می توان یکی از سه وضعیت زیر را بررسی کرد؟

- $f(n) = O(g(n))$،
- $f(n) = \Theta(g(n))$،
- $f(n) = (g(n))$.

پاسخ به این سوال **منفی** است. برای درک بهتر این موضوع، دو تابع $f(n) = n$ و $g(n) = n^{(1+sin(n))}$ را در نظر بگیرید. همانطور که می‌دانیم، توان $g(n)$ بین 0 و 2 نوسان می‌کند، بنابراین نمی‌توانیم طبق تعاریف بیان شده c و n_0 ایی پیدا کنیم که شرایط تعریف سایر نمادهای مجانبی معرفی شده را برآورده کند.

۳ رابطه بین مفاهیم پیچیدگی زمانی و نمادها

شاید در نگاه اول این‌گونه به نظر برسد که بهترین حالت معادل Ω، بدترین حالت معادل $\mathcal{O}$ و حالت متوسط معادل Θ است. با رجوع به تعاریف در خواهید یافت که نمادها در حالت کلی برای مقایسه نرخ رشد توابع نسبت به یکدیگر است، در حالی که حالت‌های بهترین، بدترین و متوسط برای زمان اجرای یک الگوریتم متناسب با نمونه ورودی‌های آن می‌باشد.

۴ مثال برای نمادهای مجانبی

۱.۴ مثال ۱

دو تابع زیر را در نظر بگیرید:

$$f(n) = \begin{cases} 2^n & n \leq 10^{10} \\ 100n & n \geq 10^{10} \end{cases},$$

$$g(n) = 2n^2 + n\ .$$

آیا $f(n) = O(g(n))$ است؟ پاسخ مثبت است. در واقع نکته پنهان در سوال این است که $f(n)$ باید به صورت $f(n) = 100n$ در نظر گرفته شود. آیا $f(n) = (g(n))$ است؟ به وضوح پاسخ منفی است.

۲.۴ مثال ۲

نشان دهید که رابطه $\log(n!) = \theta(n \log n)$ برقرار است.

پاسخ: طبق تعریف می‌توان گفت تساوی فوق هنگامی برقرار است که به طور همزمان داشته باشیم $log(n!) = (nlogn)$ و $log(n!) = O(nlogn)$ می‌باشد. عبارت $log(n!) = O(nlogn)$ به وضوح برقرار است اما برای اثبات قسمت $log(n!) = (nlogn)$ از تقریب استرلینگ استفاده می‌کنیم.

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

۳.۴ مثال ۳

فرض کنید $f(n) = O(g(n))$ باشد. آیا می‌توان گفت رابطه $2^{f(n)} = \mathcal{O}(2^{g(n)})$ برقرار است؟ پاسخ منفی است. برای درک بهتر به مثال زیر توجه کنید: اگر $f(n) = 2n$ و $g(n) = n$ بنابر فرض مسئله $f(n) = O(g(n))$ پس داریم:

$$2^{f(n)} = O(2^{g(n)}) \quad \to \quad 2^{2(n)} = O(2^n) \quad \to \quad 4^n \leq 2^n$$

که این عبارت به وضوح صحیح نیست.

۴.۴ مثال ۴

فرض کنید $f(n) = \mathcal{O}(g(n))$ باشد و $\log(g(n)) \geq 1$ و $f(n) \geq 1$ برای تمامی nهای به اندازه کافی بزرگ باشد. نشان دهید که رابطه $\log(f(n)) = \mathcal{O}(\log(g(n))$ برقرار است.

پاسخ: با بازنویسی تعریف برای رابطه $f(n) = \mathcal{O}(g(n))$ داریم:

$$\exists\, c, n_1 \quad \text{s.t.} \quad \forall\, n \geq n_1 \quad 0 \leq f(n) \leq cg(n)$$

طبق صورت سوال رابطه $1 \leq f(n) \leq cg(n)$ را داریم. اگر از طرفین لگاریتم بگیریم، خواهیم داشت:

$$0 \leq \log(f(n)) \leq \log(cg(n)) \quad \to \quad 0 \leq \log(f(n)) \leq \log c + \log(g(n)) \quad (*)$$

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

$$\exists\, d, n_2 \quad s.t \quad \forall\, n \geq n_2 \quad 0 \leq \log(f(n)) \leq d\log(g(n))$$

با توجه به رابطه (*) خواهیم داشت:

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

و در نهایت داریم: $d = \log(c) + 1$ و $n_1 = n_2$.