یک: تحلیل مجانبی الگوریتم‌ها

ساختمان داده ها و الگوریتم

مدرس: دکتر نجمه منصوری

نگارنده: سجاد هاشمیان

۱. مقدمه

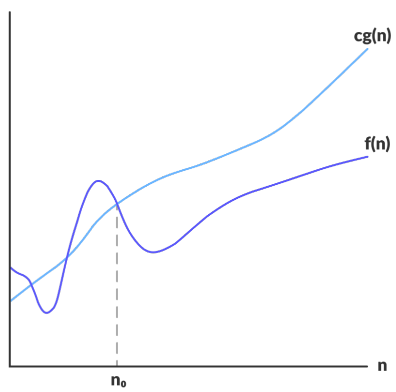
الگوریتم ابزاری است که از آن برای حل مسأله استفاده میشود. گاهی اوقات بیش از یک الگوریتم برای حل یک مسأله وجود دارد؛ بنابراین بهینه بودن ابزارها حائز اهمیت است. مهمترین پارامترهای بهینه بودن یک الگوریتم، میزان حافظه و زمانی است که برای اجرا مصرف می‌کند و این دو پارامتر، معیارهایی برای سنجش کارآیی الگوریتم هستند. در این درس به تحلیل زمانی الگوریتم‌ها می‌پردازیم. تحلیل زمانی الگوریتم معمولاً به بررسی زمان اجرا در بدترین حالت و یا حالت میانگین می‌پردازد. توجه کنید که بهترین زمان اجرای الگوریتم اهمیت چندانی ندارد، چرا که این حالت معمولاً برای ورودیهای خاص و یا با احتمال خیلی کم اتفاق می‌افتد. در علوم کامپیوتر برای اشاره به زمان اجرای الگوریتم ها از لفظ پیچیدگی استفاده می شود. به طور دقیق تر منظور از پیچیدگی، تابعی است که زمان اجرای الگوریتم را برحسب اندازه‌ی ورودی بیان می کند. در بحث تحلیل الگوریتم ها بجا درنظر گرفتن رفتار دقیق توابع، رفتار مجانبی آنها مورد بررسی قرار می‌گیرد. که در ادامه بحث به چگونگی محاسبه‌ی آن خواهیم پرداخت.

۲. نماد های مجانبی

۲.۱. نماد O

مجموعه ی توابع از مرتبه ی Oی تابع $g(n)$ با $O(g(n))$ نشان داده می شود و به صورت زیر تعریف می شود:

$$O(g(n)) = \{f(n) \,|\, \exists c > 0 \;\; and \;\; n_o \;\; .s.t \;\; \forall n > n_o, 0 \leq f(n) \leq c \times g(n)\}$$

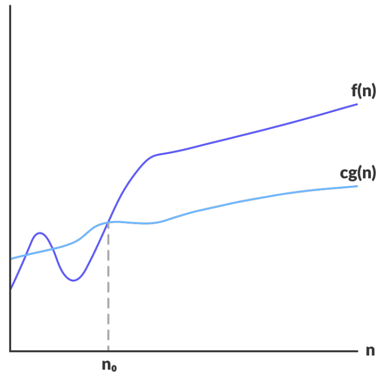


وقتی میگوییم "پیچیدگی زمانی یک الگوریتم $O(g(n))$ است"، منظورمان این است که یک تابع $f(n)$ از $O(g(n))$ وجود دارد که برای مقادیر به اندازه کافی بزرگ n، بدون توجه به این که چه ورودی ای با اندازه ی n انتخاب شده است، زمان اجرای الگوریتم برای آن ورودی از بالا با ضریب ثابت مثبتی از $g(n)$ کراندار شده است؛ یا به عبارتی زمان اجرا در بدترین حالت $O(g(n))$ است و می‌نویسیم $f(n) = O(g(n))$.

۲.۲. نماد Ω

همانطور که نماد O کران بالای مجانبی یک تابع را مشخص می کند، نماد Ω کران پایین مجانبی توابع را نشان می دهد.
مجموعه ی توابع از مرتبه ی Ωی تابع $g(n)$ با $\Omega(g(n))$ نشان داده می شود و به صورت زیر تعریف می شود:

$$\Omega(g(n)) = \{f(n) \,|\, \exists c > 0 \;\; and \;\; n_o \;\; .s.t \;\; \forall n > n_o, 0 \leq c \times g(n) \leq f(n)\}$$

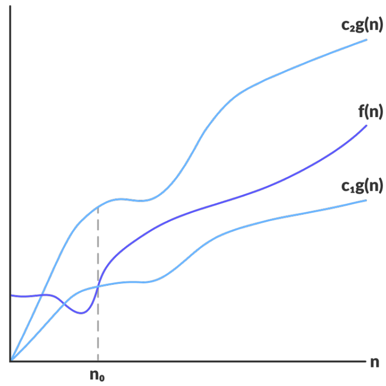


میگوییم "پیچیدگی زمانی یک الگوریتم $\Omega(g(n))$ است"، منظورمان این است که یک تابع $f(n)$ از $\Omega(g(n))$ وجود دارد که برای مقادیر به اندازه کافی بزرگ n، بدون توجه به این که چه ورودی ای با اندازه ی n انتخاب شده است، زمان اجرای الگوریتم برای آن ورودی حداقل ضریب ثابتی از $g(n)$ است؛ یا به عبارتی زمان اجرا در بهترین حالت $\Omega(g(n))$ است و می‌نویسیم $f(n) = \Omega(g(n))$.

۲.۳. نماد Θ

مجموعه‌ی توابع از مرتبه‌ی Θی تابع $g(n)$ با $\Theta(g(n))$ نشان داده می شود و به صورت زیر تعریف می شود:

$$\theta(g(n)) = \{f(n) \,|\, \exists c > 0 \;\; and \;\; n_o \;\; .s.t \;\; \forall n > n_o, c_2 \times g(n) \leq f(n) \leq c_1 \times g(n)\}$$



به عبارت دیگر، اگر ضرایب ثابت c_1 و c_2 وجود داشته باشند که برای nهای به اندازه کافی بزرگ تابع $f(n)$ بین $c_1g(n)$ و $c_2g(n)$ قرار بگیرد، تابع $f(n)$ به مجموعه ی $\Theta(g(n))$ تعلق دارد.
چون $\Theta(g(n))$ یک مجموعه است، میتوانیم بنویسیم $f(n) \in \Theta(g(n))$ تا نشان دهیم $f(n)$ عضوی از آن است، با این حال به جای آن از $f(n) \in \Theta(g(n))$ استفاده میکنیم تا مفهوم مشابه ی را نشان دهیم. میتوانیم بگوییم $g(n)$ یک کران دوطرفه مجانبی برای $f(n)$ است. همچنین، در اصلاح میگوییم $f(n)$ از مرتبهی Θی $g(n)$ است.

لم. یک تابع چندجمله‌ای، از مرتبه‌ی دقیق جمله‌ی با بزرگترین توانش است. یعنی با فرض $k < 0$ و $a_k < 0$ داریم:

$$a_k n^k + a_{k-1} n^{k-1} + \ldots = \theta(n^k)$$

قضیه. $f(n) = \theta(g(n))$ اگر و فقط اگر $f(n) = O(g(n))$ و $f(n) = \Omega(g(n))$ باشند.

۳. دانستنی‌های مفید

در این بخش چند تساوی و نامساوی را که در تحلیل الگوریتم‌ها مفید هستند، بدون اثبات ارائه می‌دهیم.

۳.۱. تصاعد حسابی

اگر $a_n = a_{n-1} + c$ یک دنباله باشد و c نیز یک مقدار ثابت باشد، داریم:

$$a_1 + a_2 + a_3 + \ldots + a_n = \frac{n(a_n + a_1)}{2}$$

به طور خاص نیز نتیجه می‌گیریم:

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

۳.۲. تصاعد هندسی

اگر $a_n = a_{n-1} \times c$ یک دنباله باشد و $c \neq 0$ نیز یک مقدار ثابت باشد، داریم:

$$a_1 + a_2 + a_3 + \ldots + a_n = a_1 \times \frac{c^n - 1}{c - 1}$$

اگر $0 < c < 1$ آنگاه حاصل جمع نامتناهی این تصاعد هندسی به صورت زیر است:

$$\sum_{i=1}^{\infty} a_i = \frac{a_1}{1 - c}$$

به طور خاص نیز نتیجه می‌گیریم:

$$1 + 2 + 4 + \ldots + 2^n = 2^{n+1} - 1$$

۳.۳. حاصل جمع مربعات

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

۳.۴. سری هارمونیک

$$H_n = \sum_{k=1}^{n} \frac{1}{k} = \ln(n) + \gamma + O(\frac{1}{n})$$

که $\gamma = 0.577...$ در آن ثابت اویلر است.

۳.۵. حاصل جمع لگاریتم ها

$$\sum_{i=1}^{n} \lfloor \log_2^i \rfloor = (n+1)\lfloor \log_2^n \rfloor - 2^{\lfloor \log_2^n \rfloor + 1} + 2 = \Theta(n \log(n))$$

۳.۶. محدود کردن حاصل جمع‌ها با انتگرال

$$\sum_{i=1}^{n} f(i) \leq \int_{1}^{n+1} f(x)\,.\,dx$$

۳.۷. تقریب استرلینگ

$$n! = \sqrt{2\pi n}\,[\frac{n}{e}]^n\,(1 + O(\frac{1}{n}))$$

به ویژه که این تقریب نتیجه می‌دهد $\log_2(n!) = \Theta(n \log n)$.

۴. خواص توابع رشد

۴.۱. خاصیت تراگذاری(Transitive)

۱. از $f(n) = \Theta(g(n))$ و $g(n) = \Theta(h(n))$ نتیجه می شود $f(n) = \Theta(h(n))$.

۲. از $f(n) = O(g(n))$ و $g(n) = O(h(n))$ نتیجه می شود $f(n) = O(h(n))$.

۳. از $f(n) = \Omega(g(n))$ و $g(n) = \Omega(h(n))$ نتیجه می شود $f(n) = \Omega(h(n))$.

۴.۲. خاصیت بازتابی(Reflexive)

۱. $f(n) = \Theta(f(n))$

۲. $f(n) = O(f(n))$

۳. $f(n) = \Omega(f(n))$

۴.۳. خاصیت تقارنی(Symmetry)

$f(n) = \Theta(g(n))$ است، اگر و فقط اگر $g(n) = \Theta(f(n))$ باشد.

۴.۴. خاصیت تقارنی ترانهاده(Transpose Symmetry)

شرط لازم و کافی برای آنکه $f(n) = O(g(n))$ باشد، آن است که $g(n) = \Omega(f(n))$ باشد.

۵. تمرین های مروری

۱. ثابت کنید که اگر $P(n)$ یک چندجمله ای از n باشد، آنگاه:

$O(\log(P(n))) = O(log(n))$

۲. ثابت کنید:

I. $n(\log_3 n)^5 = O(n^{\frac{6}{5}})$

II. $(\log_2 n)^b = O(n^b)$

۳. ثابت کنید اگر $f(n) = O(s(n))$و$g(n) = O(r(n))$ انگاه خواهیم داشت $f(n) + g(n) = O(s(n) + r(n))$.

۴. برای هرکدام، یک مثال نقض بیاورید:

I. اگر $f(n) = O(s(n))$ و $g(n) = O(r(n))$ انگاه خواهیم داشت $f(n) - g(n) = O(s(n) - r(n))$.

II. اگر $f(n) = O(s(n))$ و $g(n) = O(r(n))$ انگاه خواهیم داشت $f(n)/g(n) = O(s(n)/r(n))$.

۵. دو تابع همواره صعودی $f(n)$ و $g(n)$ را چنان بیابید که $f(n) \neq O(g(n))$ و $g(n) \neq O(f(n))$.

۶. توابع زیر را از لحاظ بزرگی (سرعت رشد در بینهایت) با هم مقایسه کنید.

f(n)	g(n)
$100n + \log(n)$	$n + (\log n)^2$
$\log(n)$	$log(n^2)$
$\frac{n^2}{\log n}$	$n \log(n^2)$
$(\log n)^{\log n}$	$\frac{n}{\log n}$
$n2^n$	3^n

۷. صحیح یا غلط بودن عبارات زیر را مشخص کنید:

I. $2n^2 + 1 = O(n^2)$

II. $\sqrt{n} = O(\log n)$

III. $\log n = O(\sqrt{n})$

IV. $n^2(1 + \sqrt{n}) = O(n^2 \log n)$

V. $3n^2 + \sqrt{n} = O(n^2)$

VI. $\sqrt{n} \log n = O(n)$