یادآوری جلسه ششم

قضیه اصلی و روش‌های تقسیم و غلبه

درّین ستوده

در جلسه قبل، در رابطه با روش‌های تقسیم و غلبه صحبت کردیم. این روش‌ها از سه بخش تشکیل شده‌اند:

۱. تقسیم :(Divide) مسئله اصلی را به a زیرمسئله با اندازه $\frac{n}{b}$ تقسیم می‌کنیم.

۲. غلبه :(Conquer) هر زیرمسئله را با استفاده از تقسیم و غلبه به صورت بازگشتی حل می‌کنیم. $a\,T(\frac{n}{b})$

۳. ترکیب :(Combine) با ترکیب زیرمسئله‌ها، مسئله اصلی را حل می‌کنیم. $f(n)$

البته دقت کنید که در حالت کلی لزومی نداره که زیرمساله‌های تولید شده اندازه یکسان داشته باشند. با این حال مثال‌هایی که در این جلسه حل می‌کنیم این خاصیت را دارند. بدین ترتیب رابطه بازگشتی $T(n) = a\,T(\frac{n}{b}) + f(n)$ حاصل می‌شود که با توجه به قضیه اصلی برای این رابطه بازگشتی، یکی از حالت‌های زیر را داریم:

$$T(n) = \begin{cases} \Theta(n^{\log_b a}) & f(n) = \mathcal{O}(n^{\log_b a - \varepsilon}) \\ \Theta(n^{\log_b a}\ \log n) & f(n) = \Theta(n^{\log_b a}) \\ \Theta(f(n)) & f(n) = \Omega(n^{\log_b a + \varepsilon}) \end{cases}$$

مثال: الگوریتم مرتب سازی ادغامی یک نمونه الگوریتم تقسیم و غلبه بوده و مراحل این الگوریتم به شرح زیر است:

۱. تقسیم: آرایه را به دو زیرآرایه با اندازه برابر تقسیم میکنیم.

۲. غلبه: هر زیرآرایه را تا حد امکان تقسیم می‌کنیم و زیرآرایه‌های حاصل را مرتب می‌کنیم. $۲T(\frac{n}{۲})$

۳. ترکیب: زیرآرایه‌ها را با رعایت ترتیب با هم ادغام میکنیم. $\Theta(n)$

$$T(n) = ۲\ T(\tfrac{n}{۲}) + \Theta(n) \xrightarrow{\text{قضیه اصلی}} T(n) = \Theta(n^{\log_۲ ۲} \log n) = \Theta(n \log n)\ \checkmark$$

مثال: آرایه A و B که هر کدام شامل n رقم که نشان دهنده یک عدد n رقمی هستند داده شده است. $A \times B$ را حساب کنید. راه حل بدیهی این مساله شبیه‌سازی ضرب عادی است که از لحاظ زمانی $\mathcal{O}(n^۲)$ است. حال سعی می‌کنیم با استفاده از روش تقسیم و غلبه این مسئله را حل کنیم:

۱. تقسیم: هرکدام از A و B را به دو قسمت برابر تقسیم می‌کنیم:

$$A = A_h \times ۱۰^{\frac{n}{۲}} + A_l \quad , \quad B = B_h \times ۱۰^{\frac{n}{۲}} + B_l$$

طبعا داریم:

$$A \times B = (A_h \times ۱۰^{\frac{n}{۲}} + A_l)(B_h \times ۱۰^{\frac{n}{۲}} + B_l) = A_hB_h \times ۱۰^n + A_hB_l \times ۱۰^{\frac{n}{۲}} + A_lB_h \times ۱۰^{\frac{n}{۲}} + A_lB_l \qquad (۱)$$

۲. غلبه: مقادیر A_hB_h ،A_lB_h ،A_hB_l و A_lB_l را به‌صورت بازگشتی به‌دست می‌آوریم. $۴T(\frac{n}{۲})$

۳. ترکیب: مقدار $A \times B$ را با استفاده از رابطه (۱) به‌دست می‌آوریم. $\mathcal{O}(n)$

زمان اجرای به این صورت خواهد بود:

$$T(n) = ۴T(\frac{n}{۲}) + \mathcal{O}(n) \xrightarrow{\text{قضیه اصلی}} T(n) = \Theta(n^{\log_۲ ۴}) = \Theta(n^۲)$$

همان‌طور که می‌بینید استفاده از روش تقسیم و غلبه فوق کمکی به بهبود زمان الگوریتم نکرد. برای بهینه‌تر کردن الگوریتم، از روشی به‌نام ضرب کاراتسوبا استفاده می‌کنیم. در این روش تنها مقادیر $(A_h + A_l)(B_h + B_l)$، A_hB_h و A_lB_l را در قسمت غلبه محاسبه کرده و به کمک این سه مقدار می‌توانیم عبارت $A_hB_l + A_lB_h$ را در $O(n)$ به‌دست آوریم:

$$(A_h + A_l)(B_h + B_l) = A_hB_h + A_lB_l + A_hB_l + A_lB_h \Rightarrow A_hB_l + A_lB_h = (A_h + A_l)(B_h + B_l) - A_hB_h - A_lB_l$$

در این‌صورت رابطه بازگشتی به صورت زیر خواهد بود:

$$T(n) = ۳\ T(\frac{n}{۲}) + \mathcal{O}(n) \xrightarrow{\text{قضیه اصلی}} T(n) = \Theta(n^{\log_۲ ۳}) = \Theta(n^{۱/۵۸})\ \checkmark$$

نکته: ممکن است اندازه $A_h + A_l$ و $B_h + B_l$ برابر با $n/۲ + ۱$ شود. آیا می‌توانید برای حل این مشکل، عملیاتی از $O(n)$ انجام دهید که یک رقم از هر دو کم شده و بخش بازگشتی دقیقا برابر با $T(n/۲)$ شود؟ لازم به تحویل دادن پاسخ نیست. به این سوال در خلوت خودتان فکر کنید!