طراحی الگوریتم‌ها

بهار ۱۴۰۰

مدرس: مسعود صدیقین

دانشکده‌ی مهندسی کامپیوتر

یادآوری جلسه اول | یادآوری

در جلسه قبل، با نمادهای $\mathcal{O}$ و Ω و θ آشنا شدیم. تعاریف این سه نماد به ازای دو تابع $f(n)$ و $g(n)$ به صورت زیر بود:

$$f(n) = \mathcal{O}(g(n)) : \quad \exists n_0, c \quad \forall n \geq n_0 \quad f(n) \leq cg(n)$$

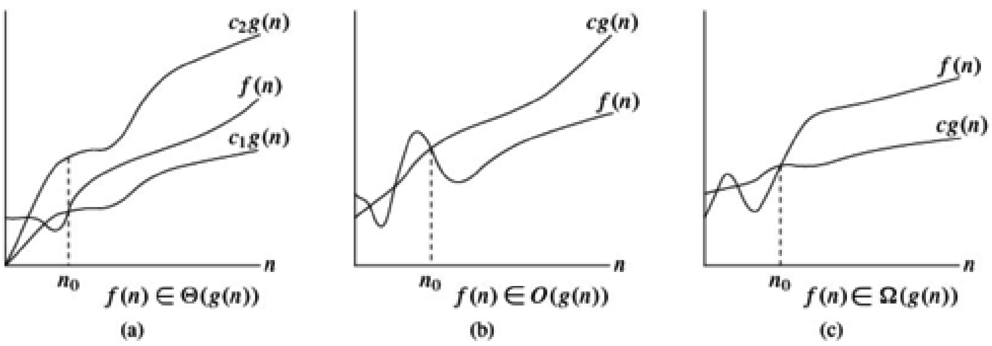
و همچنین

$$f(n) = \Omega(g(n)) : \quad \exists n_0, c \quad \forall n \geq n_0 \quad f(n) \geq cg(n)$$

و

$$f(n) = \theta(g(n)) : \quad \exists n_0, c_1, c_2 \quad \forall n \geq n_0 \quad c_1 g(n) \leq f(n) \leq c_2 g(n)$$

تصویر زیر رابطه تابع $f(n)$ و $g(n)$ را در تعریف $\mathcal{O}, \Omega, \theta$ نشان می‌دهد:



همچنین با نمادهای o و ω که به ترتیب شکل اکید نمادهای $\mathcal{O}$ و Ω هستند نیز آشنا شدیم. سپس برای هر کدام از نمادهای بالا تعدادی نمونه حل شد. برای مثال نشان دادیم به ازای هر $\epsilon > 0$ داریم: $\log n = o(n^{\epsilon})$. همچنین از ترم قبل به یاد دارید که برای تحلیل روابط بازگشتی از قضیه اصلی استفاده می‌کردیم. فرم کلی قضیه اصلی به این صورت است که اگر داشته باشیم:

$$T(n) = aT(n/b) + f(n),$$

در این صورت، هر کدام از حالت‌های زیر برقرار باشد، می‌توان فرم صریح $T(n)$ را به دست آورد:

۱. اگر به ازای یک $\epsilon > 0$ داشته باشیم $f(n) = \mathcal{O}(n^{\log_b^a - \epsilon})$ آنگاه $T(n) = \theta(n^{\log_b^a})$.

۲. اگر $f(n) = \theta(n^{\log_b^a})$ آنگاه $T(n) = \theta(n^{log_b^a} \log n)$.

۳. اگر به ازای یک $\epsilon > 0$ داشته باشیم $f(n) = \Omega(n^{\log_b^a + \epsilon})$ آنگاه $T(n) = \theta(f(n))$.

پرسش رابطه بازگشتی $T(n) = 7T(n/2) + n^2$ را با استفاده از قضیه اصلی حل کنید. با فرض این که جواب این رابطه $T(n) = \theta(n^x)$ است، مقدار x را تعیین کنید.

پاسخ های خود را می توانید تا قبل از شروع کلاس به این لینک ارسال کنید.