دانشکده علوم ریاضی و آمار

مدرس: دکتر مجتبی رفیعی نیمسال اول ۱۴۰۰-۱۴۰۱

ساختمان داده‌ها و الگوریتم‌ها

جلسه ۲۱

نگارنده: شیوا حسینی زاده

۲۵ آبان ۱۴۰۰

فهرست مطالب

۱ اثبات قضیه اصلی-حالت سوم ۱

۲ سقف و کف در توابع بازگشتی ۲

۳ مقدمه ای بر ساختمان داده ۴

۱ اثبات قضیه اصلی-حالت سوم

$$T(n) = aT(\frac{n}{b}) + f(n) \qquad a \geq 1,\ b > 1,\ c = \log a_b$$

$$if \qquad f(n) = \Omega(n^{c+\epsilon}) \qquad and \qquad af(\frac{n}{b}) \leq hf(n) \qquad where \qquad 0 < h < 1 \qquad then \qquad T(n) = \Theta(f(n))$$

و این رابطه برای هر n به اندازه کافی بزرگ برقرار است. طبق رابطه ای که برای T(n) از روش درخت بازگشت بدست آوردیم،داریم:

$$T(n) = n^c \cdot T(1) + \sum_{i=0}^{(\log_b n)-1} a^i f(\frac{n}{b^i})$$

طبق شرایط حالت سوم داریم:

$$af(\frac{n}{b}) \leq hf(n)$$

دو طرف را بر a تقسیم می‌کنیم:

$$f(\frac{n}{b}) \leq \frac{h}{a}f(n)$$

حال ورودی f را $\frac{n}{b}$ می‌گیریم:

$$f(\frac{n}{b^2}) \leq \frac{h}{a}f(\frac{n}{b})$$

داریم:

$$f(\frac{n}{b^2}) \leq \frac{h}{a} \times \frac{h}{a} \times f(n) = \frac{h^2}{a^2}f(n)$$

$$\vdots$$

برای k داریم:

$$f(\frac{n}{b^k}) \leq \frac{h^k}{a^k}f(n) \longrightarrow a^k f(\frac{n}{b^k}) \leq h^k f(n)$$

با استفاده از رابطه ی بالا داریم:

$$\sum_{i=0}^{(\log_b n)-1} a^i f(\frac{n}{b^i}) \leq \sum_{i=0}^{(\log_b n)-1} h^i f(n) = f(n) \sum_{i=0}^{(\log_b n)-1} h^i$$

و از آنجایی که $0 < h < 1$ داریم:

$$\sum_{i=0}^{(\log_b n)-1} a^i f(\frac{n}{b^i}) \leq f(n) \sum_{i=0}^{(\log_b n)-1} h^i \leq f(n)\sum_{i=0}^{\infty}h^i = f(n)(\frac{1}{1-h}) = \Theta(f(n))$$

۲ سقف و کف در توابع بازگشتی

یادآوری: برای الگوریتم مرتب سازی ادغامی، پیچیدگی زمانی آن به طور دقیق به صورت زیر تعریف شد:

$$T(n) = T(\lfloor\frac{n}{2}\rfloor) + T(\lceil\frac{n}{2}\rceil) + n$$

سوال:چرا در محاسبات خود،سقف و کف را حذف کردیم؟ چون دراین مثال ،اختلاف خیلی کم است و یک واحد کم و یا زیاد شدن معمولا تاثیری در جواب نهایی ندارد.
نکته: چون$T(n)$ چندجمله ای بود این کار ممکن بود و نمی توان همواره این فرض نادیده گرفتن را داشت مثلا:

$$a^{\lfloor\frac{n}{2}\rfloor} = \begin{cases} a^{\frac{n}{2}} \\ a^{\frac{n}{1}+1} \end{cases},$$

در $T(n)$ تاثیری ندارد،بنابراین از آنجایی که $T(n)$ برای الگوریتم مرتب سازی ادغامی چندجمله ای بود:

$$T(n) = T(\lfloor\frac{n}{2}\rfloor) + T(\lceil\frac{n}{2}\rceil) + n \leq 2T(\lceil\frac{n}{2}\rceil) + n$$

و برای رهایی از سقف :

$$T(n) \leq 2T(\frac{n}{2}+1) + n$$

همچنین برای حل از طریق قضیه اصلی (۱) را هم نادیده گرفتیم:

$$T'(n) = 2T(\frac{n}{2}) + n$$

درنهایت قضیه اصلی می تواند یک حدس مناسب برای استفاده از روش حدس و استقرا باشد.
جمع بندی: اگر $T(n)$چندجمله ای باشد **سقف** و **کف** و عدد ۱ ($\lceil \frac{n}{2} \rceil \leq \frac{n}{2} + 1$) تاثیر چندانی ندارد.
حل رابطه ی بازگشتی زیر به کمک درخت بازگشت:

$$T(n) = aT(\lceil \frac{n}{b} \rceil) + f(n)$$

یادآوری برای رابطه ی بازگشتی: $T(n) = aT(\frac{n}{b}) + f(n)$

$$\begin{array}{ccccccc}
 & & & T(n) & & & \\
 & & T(\frac{n}{b}) & T(\frac{n}{b}) & ... & T(\frac{n}{b}) \rightarrow & f(n) \\
T(\frac{n}{b^2}) & T(\frac{n}{b^2}) & ... & T(\frac{n}{b^2}) & \cdots & \rightarrow & af(\frac{n}{b}) \\
\vdots & \vdots & & \vdots & & & \vdots \\
T(1) & \cdots & & \cdots & & \cdots \rightarrow & T(1)
\end{array}$$

تا کجا باید پیش برویم:

$$\frac{n}{b^k} = 1 \longrightarrow k = \log_b n$$

تعداد نود ها در سطح k:

$$a^k = a^{\log_b n} = n^{\log_b a} = n^c \qquad where\ c = \log_b a$$

$$T(n) = n^c T(1) + a^{(\log_b n)-1} f(1) + \sum_{i=1}^{k-1} a^{i-1} f(\frac{n}{b^{i-1}})$$

$$\sum_{i=1}^{k-1} a^{i-1} f(\frac{n}{b^{i-1}}) = \sum_{i=0}^{k-2} a^i f(\frac{n}{b^i})$$

حال تابع $T(n) = 2T(\lceil \frac{n}{b} \rceil) + f(n)$ را در نظر بگیرید برای سطوح مختلف درخت بازگشت داریم:
سطح یک :

$$f(n) = f(n_0)$$

سطح دو :

$$f(\lceil \frac{n}{b} \rceil) = f(n_1)$$

سطح سه :

$$f(\lceil \frac{\lceil \frac{n}{b} \rceil}{b} \rceil) = f(n_2)$$

می توانیم ورودی های f را به صورت زیر تعریف کنیم:

$$n_k = \begin{cases} n & if \quad k=0 \\ \lceil \frac{n_k - 1}{b} \rceil & if \quad k>0 \end{cases},$$

حذف سقف برای n_k ها:

$$\lceil x \rceil \leq x+1$$

$$n_0 = n$$

$$n_1 = \lceil \frac{n_0}{b} \rceil = \lceil \frac{n}{b} \rceil \leq \frac{n}{b} + 1$$

$$n_2 = \lceil \frac{n_1}{b} \rceil = \lceil \frac{\lceil \frac{n}{b} \rceil}{b} \rceil \leq \frac{\frac{n+b}{b}}{b} + 1 = \frac{n}{b^2} + \frac{1}{b} + 1$$

$$\vdots$$

$$n_k \leq \frac{n}{b^k} + \sum_{i=0}^{k-1} \frac{1}{b^i} < \frac{n}{b^k} + \sum_{i=0}^{\infty} \frac{1}{b^i}$$

چون $b > 1$ است داریم:

$$= \frac{n}{b^k} + \frac{b}{b-1}$$

پس داریم:

$$T(n) \leq \Theta(n^c) + \sum_{k=0}^{(\log_b n)-1} a^k f(n)$$

۳ مقدمه ای بر ساختمان داده

مجموعه ها یک مفهوم پایه در ریاضیات و علوم کامپیوتر هستند. با این حال به طور معمول مجموعه ها در ریاضیات به طور ثابت در نظر گرفته میشوند اما در علوم کامپیوتر امکان بروز رسانی (حذف و اضافه کردن) عناصر آنها نیز همواره مورد توجه بوده است و اصطلاحا با مجموعه های پویا سر و کار داریم. در ادامه درس بر روی یک سری از تکنیک های پایه برای نمایش مجموعه های پویای متناهی به عنوان بلاک اصلی ذخیره و بازیابی اطلاعات در سیستم های کامپیوتری خواهیم پرداخت برخی کاربردهای واقعی که از مجوعه های پویای متناهی بهره می گیرند:
۱ **دیکشنری:** که شامل:

- یک مجموعه از کلید واژاه هاست،
- یک سری از عملیات مثل درج کلیدواژه،حذف کلید واژه موجود وتست عضویت.

۲ **صف الویت-مینیمم:** که شامل:

- یک مجموعه از مقادیر،
- یک سری از عملیات مثل درج عنصر و استخراج کوچکترین عضو از مجموعه.