$$T(n) = aT(n/b) + f(n), \quad a \geq 1, b > 1, C = \log_b a$$

حالت ۳ :

$$\text{if} \quad f(n) = \Omega(n^{c+\varepsilon}) \quad \text{and} \quad af(n/b) \leq h\, f(n), \text{ wher } 0 < h < 1$$

و این رابطه برای هر n به اندازه کافی بزرگ برقرار است.

$$\text{then} \quad T(n) = \Theta(f(n)).$$

طبق رابطه‌ای که برای $T(n)$ از روش‌های جایگذاری و درخت بازگشت بدست آوردیم، داریم:

$$T(n) = \underbrace{n^c * T(1)}_{\Theta(n^c)} + \underbrace{\sum_{i=\phi}^{(\log_b n)-1} a^i f\left(\frac{n}{b^i}\right)}_{A^*}$$

طبق شرایط حالت سوم داریم:

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

دو طرف را بر a تقسیم کنیم

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

ورودی f را n/b تغییر دهیم.

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

داریم

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

$$\vdots$$

برای k داریم

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

با استفاده از رابطه بالا داریم:

$$A^* \leq \sum_{i=0}^{(\log_b n)-1} h^i f(n) = f(n) \sum_{i=0}^{(\log_b n)-1} h^i$$

ازآنجاییکه $1>h>0$ داریم :

$$A^{*} \leqslant f(n) \sum_{i=0}^{(\log_b n)-1} h^i \leqslant f(n) \sum_{i=0}^{\infty} h^i = f(n) \left(\frac{1}{1-h}\right) = \theta(f(n))$$

سقف و کف در توابع بازگشتی:

یادآوری: برای الگوریتم مرتب‌سازی ادغامی، پیچیدگی زمانی آن به‌طور دقیق به‌صورت زیر تعریف شد:

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

سؤال: چرا در محاسبات خود، سقف و کف را حذف کردیم؟ چون در این مثال، اختلاف خیلی کم است و یکی کم و زیاد شدن معمولاً تأثیر در جواب نهایی ندارد.

نکته: چون $T(n)$ چندجمله‌ای بود این کار ممکن بود و نمی‌توان همواره این فرض نادیده گرفتن را داشت. مثلاً

$$a^{\lfloor n/2 \rfloor} \rightarrow a^{n/2}$$
$$\quad \rightarrow a^{n/2+1}$$

در $T(n)$ تأثیرگذار است.

★ بنابراین از آنجاییکه $T(n)$ برای الگوریتم مرتب‌سازی ادغامی، چندجمله‌ای بود:

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

* برای رهایی از سقف:

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

* همچنین برای حل از طریق قضیه اصلی ⌊⌋ را هم نادیده گرفتیم:

$$T(n) = 2T(n/2) + n$$

در نهایت تفصیل عملی می‌توانند یک حدس مناسب برای استفاده از روش حدس و استقرا باشد.

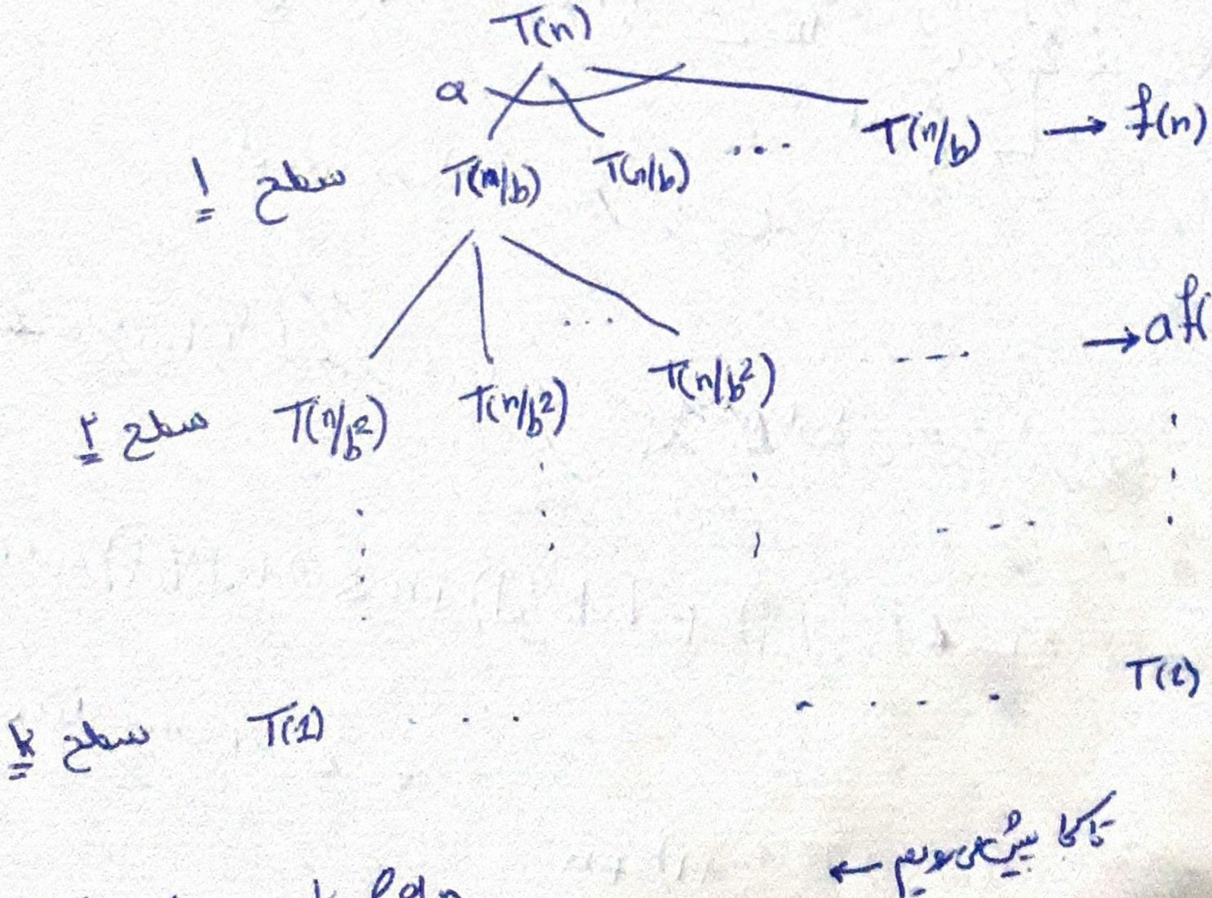
جمع بندی: اگر $T(n)$ خیر معمار باشد، سقف و کف و عدد ۱ تأثیر چندانی ندارد

$(\lceil \frac{n}{2} \rceil < \frac{n}{2}+1)$

حل رابطه بازگشتی زیر به کمک درخت بازگشت:

$$T(n) = aT(\lceil n/b \rceil) + f(n)$$

یادآوری برای رابطه بازگشتی: $T(n)=aT(n/b)+f(n)$



تا کجا پیش می‌رویم ←

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

تعداد نودها در سطح k ←

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

$$T(n) = \underbrace{n^c T(1) + a^{(\log_b n)-1} f(1)}_{\text{سطح آخر}} + \underbrace{\sum_{i=1}^{k-1} a^{i-1} f\left(\frac{n}{b^{i-1}}\right)}_{A^*}$$

$$A^* = \sum_{i=0}^{k-2} a^i f\left(\frac{n}{b^i}\right)$$

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

سطح ۱ $\longrightarrow f(n) = f(n_0)$

سطح ۲ $\longrightarrow f(\lceil n/b \rceil) = f(n_1)$

سطح ۳ $\longrightarrow f(\lceil \lceil n/b \rceil / b \rceil) = f(n_2)$

$\vdots$

می‌توانیم ورودی های f را به صورت زیر تعریف کنیم:

$$n_k = \begin{cases} n & \text{if } k=0 \\ \lceil n_{k-1}/b \rceil & \text{if } k>0 \end{cases}$$

حذف سقف برای n_k ها: $\lceil x \rceil \leq x+1$

$$n_0 = n$$

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

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

$\vdots$

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

چون $b>1$ است داریم

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

پس داریم :

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

ساختمان داده:

مجموعه ها یک مفهوم پایه در ریاضیات و علوم کامپیوتر هستند. با اینحال به طور معمول مجموعه ها در ریاضیات به طور ثابت درنظر گرفته می شوند اما در علوم کامپیوتر امکان بروزرسانی (حذف و اضافه کردن) عناصر آنها نیز همواره مورد توجه بوده است و اصطلاحاً با مجموعه های پویا سروکار داریم.

در ادامه درس، بر روی یکسری 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 مجموعه های پویای متناهی بهره می گیرند:

۱. دیکشنری: که شامل
- یک مجموعه از کلید واژه هاست،
- یک سری از عملیات مثل:
  - درج کلید واژه جدید،
  - حذف کلید واژه موجود،
  - تست عضویت.

۲. صف اولویت - مینیمم: که شامل
- یک مجموعه از مقادیر است،
- یکسری از عملیات مثل
  - درج عنصر،
  - استخراج کوچکترین عضو از مجموعه.