دانشکده علوم ریاضی و آمار

مدرس: دکتر مجتبی رفیعی نیمسال اول ۱۴۰۰-۱۴۰۱

ساختمان داده‌ها و الگوریتم‌ها

جلسه ۱۸: ساختمان داده و الگوریتم‌ها

نگارنده: عاطفه قوقه‌ای

۸ آبان ۱۴۰۰

فهرست مطالب

۱ حل روابط بازگشتی-روش جایگذاری

مثال برای روش جایگذاری: تابع بازگشتی زیر را که در یک فرم کلی داده شده است در نظر بگیرید:

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

گام اول جایگذاری:

$$\begin{aligned} T(n) &= aT(\frac{n}{b}) + f(n) \\ &= a(aT(\frac{n}{b^2}) + F(\frac{n}{b})) + f(n) \\ &= a^2T(\frac{n}{b^2}) + af(\frac{n}{b}) + f(n) \end{aligned}$$

گام دوم جایگذاری :

$$T(n) = a^2T(\frac{n}{b^2}) + af(\frac{n}{b})) + f(n)$$

$$= a^2(aT(\frac{n}{b^3}) + F(\frac{n}{b^2})) + aF(\frac{n}{b}) + F(n)$$

$$= a^3T(\frac{n}{b^3}) + a^2F(\frac{n}{b^2})) + aF(\frac{n}{b}) + F(n)$$

گام K–ام جایگذاری

$$T(n) = a^kT(\frac{n}{b^k}) + \sum_{i=0}^{k-1} a^iF(\frac{n}{b^i})$$

حال سوالی که پیش می‌آید این است که تا کجا باید k را پیش ببریم؟ تا جایی که $T(\frac{n}{b^k})$ به حالت پایه که برابر یک است برسد.

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

پس رابطه کلی زیر حاصل می‌شود:

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

از آن جایی که $n^{\log_b a} = a^{\log_b n}$ است می‌توان عبارت بالا را به صورت زیربازنویسی کرد:

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

رابطه‌ی بازگشتی ما دیگر براساس جملات قبلی نیست بلکه براساس اندازه ورودی آن است. حال در ادامه قصد داریم دو حالت خاص از فرم کلی رابطه بالا را در نظر گرفته و حل کنیم:

ابتدا فرض کنید که $f(n) = C$ باشد بنابراین داریم:

$$T(n) = n^{\log_b a}T(1) + \sum_{i=0}^{(\log_b n)-1} Ca^i$$

$$= \Theta(n^{\log_b a}) + \Theta(n^{\log_b a}) = \Theta(n^{\log_b a})$$

$$\underbrace{n^{\log_b a}T(1)} \qquad \longmapsto \qquad \Theta(n^{\log_b a})$$

$$\underbrace{\sum_{i=0}^{(\log_b n)-1} Ca^i} \qquad = C\sum_{i=0}^{(\log_b n)-1} a^i = C\frac{a^{(\log_b n)-1+1}}{a-1}$$

$$= C\frac{n^{\log_b a}-1}{a-1} = \Theta(n^{\log_b a})$$

نکته: برای هر عدد حقیقی چون $a \neq 1$ باشد داریم:

$$\sum_{i=0}^{u} a^i = \frac{a^{u+1}}{a-1}$$

فرض کنید $f(n) = n$ باشد بنابراین داریم:

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

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

$$\sum a^i * \frac{n}{b^i} = n \sum_{i=0}^{(\log_b n)-1} (\frac{a}{b})^i = n * \frac{(\frac{a}{b})^{\log_b n} - 1}{\frac{a}{b} - 1}$$

$$(\frac{a}{b})^{\log_b n} = n^{\log_b \frac{a}{b}} = n^(\log_b a - \log_b b) = n^{\log_b a - 1} = O(n^{\log_b a})$$

نکته: اگر روابط بازگشتی پیچیده‌تر شود مثلا تابع بازگشتی براساس دو تابع کوچکتر تعریف شود:

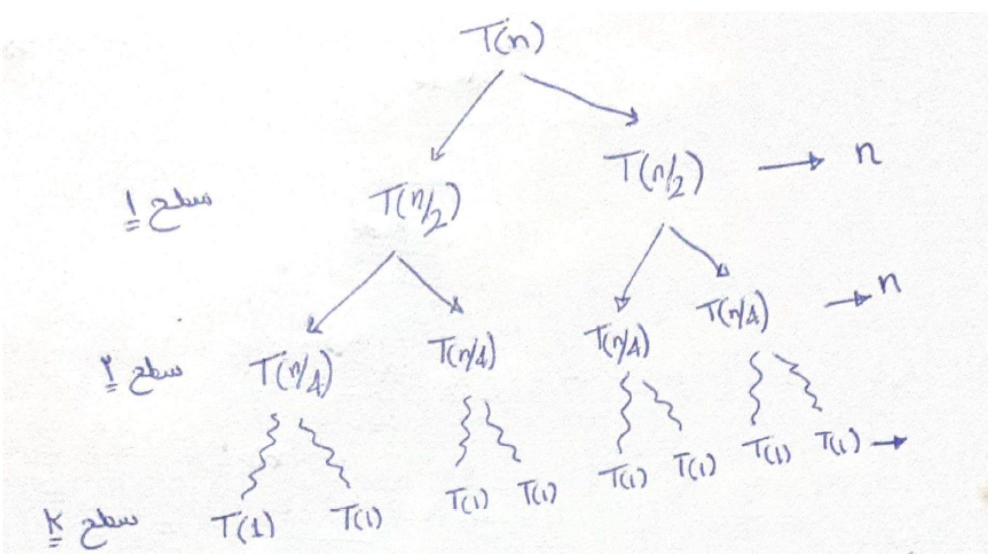
$$T(n) = 2T(\frac{n}{2}) + 3T(\frac{2n}{5}) + n$$

در این حالت روش جایگذاری آزاردهنده بوده و ممکن است نتوان از این روش بهره گرفت.

۲ حل روابط بازگشتی-روش درخت بازگشت

می‌توان برای حل روابط بازگشتی درخت آن را ترسیم کرد که به آن درخت بازگشت گفته می‌شود.
مثال۱.

$$T(n) = 2T(\frac{n}{2}) + n$$



حال سوال این است که تا کجا پیش برویم؟

$$\frac{n}{2^k} = 1 \qquad \longmapsto \qquad n = 2^k \qquad \longmapsto \qquad \log_2 n = k$$

تعداد نودها در سطح k-ام:

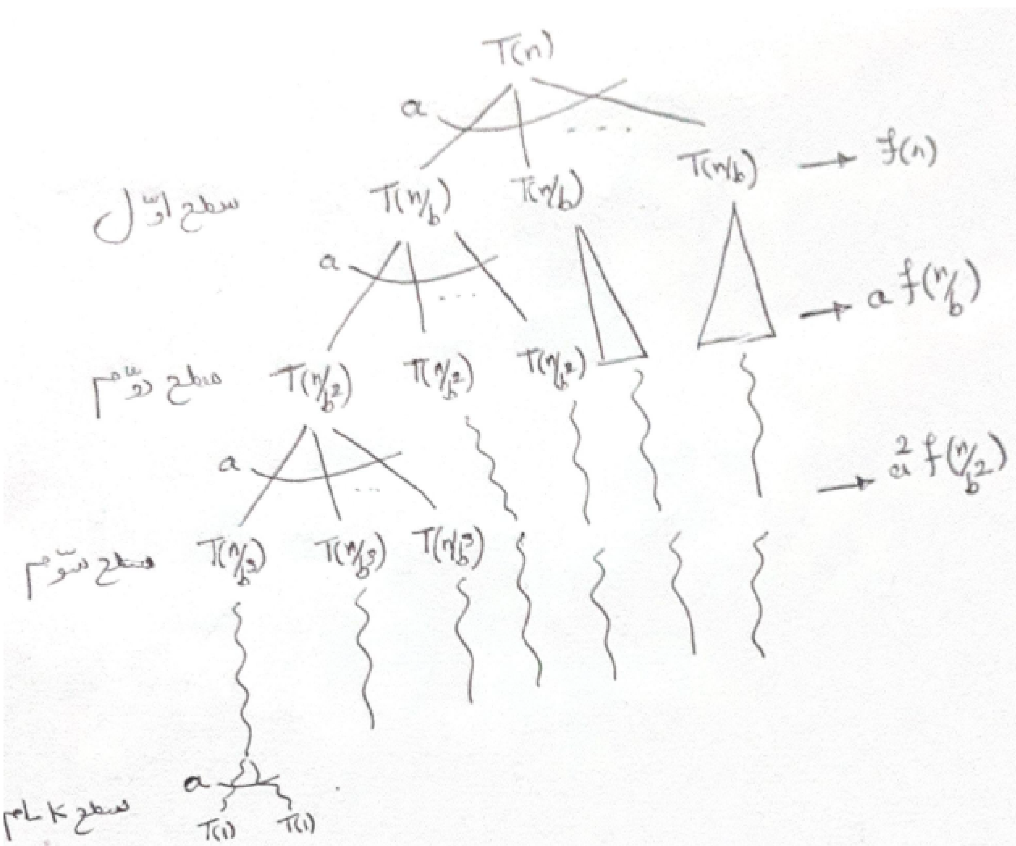
$$2^k = 2^{\log_2 n} = n$$

$$x = nT(1) + \frac{n}{2} * 2$$

$$T(n) = nT(1)+n+\sum_{i=1}^{k-1} n = nT(1)+n+n(k-1) = nT(1)+n+n(\log_2 n-1) = nT(1)+n\log_2 n = \Theta(n\log_b n)$$

مثال۲.

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



تا کجا پیش می‌رویم؟

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

تعداد نودها در سطح kام:

$$a^k = a^{\log_b n} = n^{\log_b a}$$

تابع درF مرحله :k

$$f(\frac{n}{b^k}) = f(\frac{n}{b^{\log_b n}}) + f(\frac{n}{n}) = f(1)$$

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