یک: روابط بازگشتی

ساختمان داده ها و الگوریتم

مدرس: دکتر نجمه منصوری

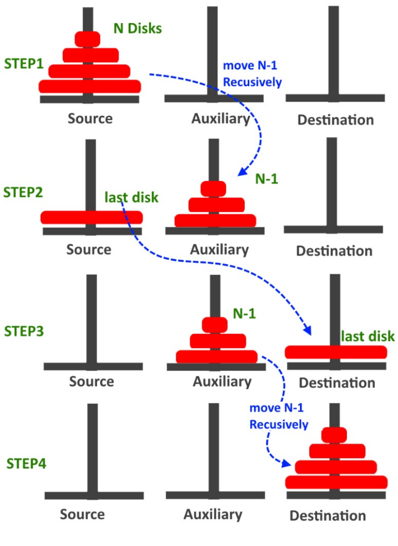
نگارنده: سجاد هاشمیان

۱. مقدمه

پیدا کردن کران مجانبی توابع معمولا با پیچیدگی همراه است و راه حل آسانی ندارد، هنگامی که روابط بازگشتی داشته باشیم این کار سخت تر هم میشود. برای محاسبه‌ی زمان اجرای الگوریتم هایی، لازم است که کران مجانبی روابط بازگشتی آنها رو پیدا کنیم. در این بخش تکنیک‌هایی برای پیدا کردن کران های روابط بازگشتی معرفی میکنیم و در انتها قضیه ی اصلی عنوان میشود که در پیدا کردن کران مجانبی طیف وسیعی از توابع کاربرد دارد.

۲. طرح چند مسئله

۲.۱. برج هانوی



برج هانوی یک مسأله معروف الگوریتمی است، که پیشینه‌ی تاریخی بسیار طولانی دارد.اولین بار این مسأله در یک معبد شکل گرفت. سه برج الماس در این معبد وجود داشت که در اولین برج، ۶۴ حلقه از کوچک به بزرگ مرتب شده بودند. کائنات قصد داشتند تا این حلقه ها را از روی این برج، به برج دیگر منتقل کنند و معتقد بودند با انجام این کار، عمر دنیا به پایان خواهد رسید.

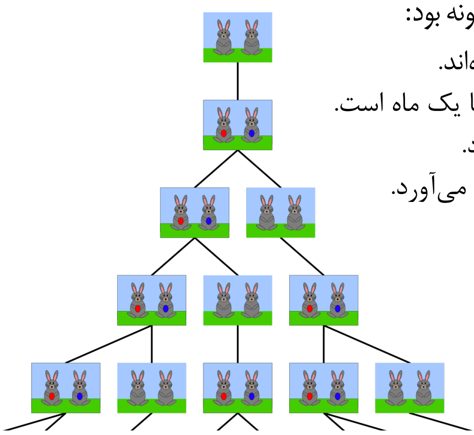
این مسئله به این صورت مطرح میشود که ما یک تخته به همراه سه میله عمودی و n دیسک با قطرهای متمایز داریم که دیسک‌ها به ترتیب طول قطرشان، روی یک میله قرار دارند. میخواهیم حداقل تعداد حرکات برای انتقال این دیسک‌ها را از یک میله به میله دیگر، به شرط آن که روی هیچ میله‌ای دیسکی روی دیسکی با قطر کمتر قرار نگیرد را محاسبه کنیم.

با توجه به توضیحات رابطه بازگشتی آن برابر است با $H(n) = 2H(n-1) + 1$ که نتیجه میدهد برای محاسبه تعداد حرکات در کل رابطه $H(n) = 2^n - 1$ را داشته باشیم.

البته در طی دوران برای این مسئله تعمیم ها و صورت های دیگری نیز بیان شده، که پیشنهاد می‌شود برای آشنا شدن با آن‌ها به اینجا مراجعه کنید.

۲.۲. دنباله فیبوناچی

فیبوناچی در سال ۱۲۰۲ به مسئله عجیبی علاقه‌مند شد. او می‌خواست بداند اگر یک جفت خرگوش نر و ماده داشته باشد و رفتاری برای زاد و ولد آن‌ها تعریف کند در نهایت نتیجه چگونه خواهد شد. فرضیات اینگونه بود:



۱. شما یک جفت خرگوش نر و ماده دارید که همین الآن به‌دنیا آمده‌اند.
۲. خرگوش‌ها پس از یک ماه بالغ می‌شوند. دوران بارداری خرگوش‌ها یک ماه است.
۳. هنگامی که خرگوش ماده به سن بلوغ می‌رسد حتماً باردار می‌شود.
۴. در هر بار بارداری خرگوش ماده یک خرگوش نر و یک ماده به‌دنیا می‌آورد.
۵. خرگوش‌ها هرگز نمی‌میرند.

حساب کنید پس از n ماه چند جفت از این نوع خرگوش خواهیم داشت؟

$$F(n) = \begin{cases} 1 & n = 1 \\ 1 & n = 2 \\ F(n-1) + F(n-2) & otherwise \end{cases}$$

به مانند هر مسئله دیگری اعداد فیبوناچی هم در تحقیقات زیادی آمده اند که برای مطالعه بیشتر می‌توانید به اینجا مراجعه کنید.

۲.۳. دومینو

فرض کنید یک مستطیل $2 \times n$ داریم که $n \in \mathbb{Z}$ است. میخواهیم این مستطیل را با مقواهایی مستطیل شکل به ابعاد 1×2، به نام دومینو بپوشانیم. به چند طریق مستطیل با دومینوها کاملاً پوشانده میشوند (دومینوها نباید روی هم قرارگیرند).

$$d(n) = \begin{cases} 1 & n = 1 \\ 2 & n = 2 \\ d(n-1) + d(n-2) & otherwise \end{cases}$$

در واقع این مسئله به سادگی قابل تبدیل به دنباله فیبوناچی است و رابطه برای آن صدق می‌کند. این نسخه از مسئله گاها به عنوان مسئله نربان مطرح می‌شود. برای تعمیم مسئله می‌توانید اندازه مستطیل یا دومینوها را تغییر دهید(البته باز هم با فیبوناچی در رابطه است!).

۳. حدس و استقرا

یکی از روش‌هایی که میتوانیم برای حل این روابط استفاده کنیم، این است که ابتدا فرم کلی جواب را حدس بزنیم، سپس با کمک استقرا مقادیر ثابت را پیدا کنیم و نشان دهیم که جوابمان درست است. این روش، روش خوبی برای حل روابط بازگشتی است، ولی **باید بتوانیم در ابتدا فرم جواب را حدس بزنیم.**

مثال. کران دوطرفه مجانبی تابع $T(n) = 2T(\frac{n}{2}) + \Theta(n)$ چیست؟

حدس میزنیم که کران دوطرفه مجانبی $\Theta(n \log n)$ باشد. برای اثبات باید نشان دهیم:

$$\begin{cases} I) & T(n) = O(n \log n) \\ II) & T(n) = \Omega(n \log n) \end{cases}$$

ابتدا (I) را اثبات میکنیم. فرض کنید c ضریب ثابتی باشد که به ازای آن رابطه‌ی $T(n) \leq 2T(\frac{n}{2}) + cn$ برای n های به اندازه‌ی کافی بزرگ برقرار است. فرض استقرا را به صورت زیر در نظر میگیریم:

"ضریب ثابت d وجود دارد به طوری که برای n های به اندازه کافی بزرگ $T(n) \leq dn \log n$"

حال با دنبال کردن اثبات استقرا شرایط لازم برای وجود چنین dای را تعیین میکنیم:

$$\begin{aligned} T(\frac{n}{2}) \leq d\frac{n}{2}\log\frac{n}{2} \Longrightarrow T(n) &\leq 2T(\frac{n}{2}) + cn \\ &\leq 2(d\frac{n}{2}\log\frac{n}{2}) + cn \\ &= dn\log n - dn + cn \end{aligned}$$

رابطه‌ی فوق نشان میدهد که اگر $d \geq c$ انتخاب شود، فرض $T(n) \leq dn \log n$ صحیح است. بنابراین (I) برقرار است.

حال (II) را اثبات میکنیم. فرض کنید c ضریب ثابتی باشد که به ازای آن رابطه‌ی $T(n) \geq 2T(\frac{n}{2}) + cn$ برای n های به اندازه‌ی کافی بزرگ برقرار است. فرض استقرا را به صورت زیر در نظر میگیریم:

"ضریب ثابت d وجود دارد به طوری که برای n های به اندازه کافی بزرگ $T(n) \geq dn \log n$"

حال با دنبال کردن اثبات استقرا شرایط لازم برای وجود چنین dای را تعیین میکنیم:

$$\begin{aligned} T(\frac{n}{2}) \geq d\frac{n}{2}\log\frac{n}{2} \Longrightarrow T(n) &\geq 2T(\frac{n}{2}) + cn \\ &\geq 2(d\frac{n}{2}\log\frac{n}{2}) + cn \\ &= dn\log n - dn + cn \end{aligned}$$

رابطه‌ی فوق نشان میدهد که اگر $d \leq c$ انتخاب شود، فرض $T(n) \geq dn \log n$ صحیح است. بنابراین (II) هم برقرار است.

مثال. کران بالای مجانبی تابع $T(n) = 4T(\frac{n}{4}) + n$ چیست؟

به اشتباه حدس میزنیم که کران بالای مجانبی $O(n)$ باشد. بنابراین فرض ناصحیح استقرا به صورت زیر است:

"ضریب ثابت c وجود دارد به طوری که برای n های به اندازه کافی بزرگ داریم $T(n) \leq cn$"

حال ببینیم چه مشکلی در روند اثبات استقرا پیش می‌آید.

پایه: با توجه به این که $T(1) = O(1)$ نیازی به چک کردن پایه استقرا نیست.

گام : فرض کنید به ازای همه‌ی n' هایی که $n' < n$ از جمله $n' = \frac{n}{4}$ فرض استقرا برقرار باشد. حال داریم:

$$T(n') \leq cn' \Longrightarrow T(\frac{n}{4}) \leq c\frac{n}{4} \Longrightarrow T(n) \leq 4(c\frac{n}{4}) + n = cn + n = (c+1)n$$

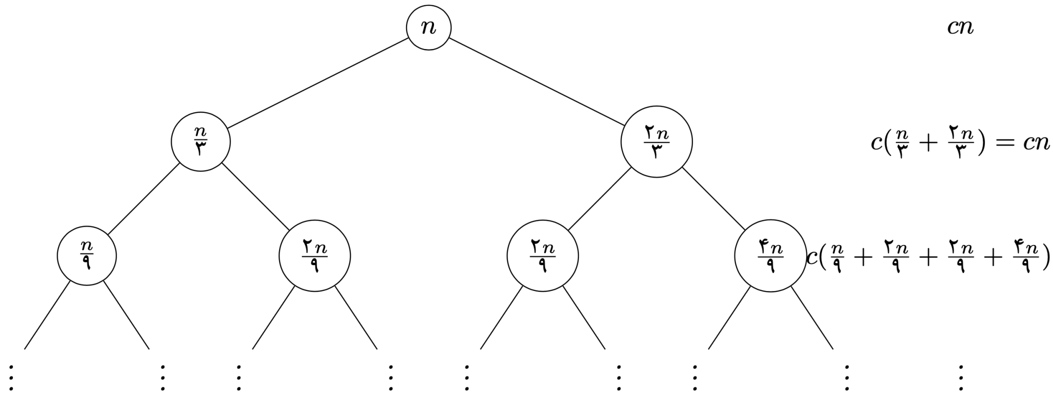
چون به فرض استقرا نرسیدیم، استدلال غلط است، پس نمیتوانیم ادعا کنیم $T(n) = O(n)$ است.

۴. درخت بازگشت

در اینجا استفاده از درخت بازگشت (درخت محاسبه) را به عنوان یک روش کارآمد برای حدس زدن جواب روابط بازگشتی معرفی میکنیم. ساختار این درخت بدین شکل است که هر گره نمایندهی یک زیر مساله از مسئلهی اصلی است و فرزندانش نشان‌دهندهی این که به صورت بازگشتی، مسئله به زیر مسئله‌هایی با چه اندازهای میشکند. برای یافتن هزینهی حل بازگشتی مسئله، در درخت پایین میرویم تا به زیر مسئلهای برسیم که هزینه آن را میدانیم(حالات پایه) و پس از آن، هزینه هر گره مجموع هزینه‌های فرزندانش به علاوهی هزینه شکستن آن به زیر مسائل (که اغلب ثابت است) میشود. با استفاده از درخت بازگشت میتوان اغلب حدس‌های متناسب‌تری زد.

مثال. کران دوطرفه مجانبی تابع $T(n) = T(\frac{n}{3}) + T(\frac{2n}{3}) + \Theta(n)$ چیست؟

با استفاده از درخت بازگشت، سعی میکنیم حدس مناسبی برای کران دوطرفه مجانبی تابع $T(n)$ بزنیم. فرض کنید c ضریب ثابتی باشد که برای nهای به اندازه کافی بزرگ رابطه $T(n) \leq T(\frac{n}{3}) + T(\frac{2n}{3}) + \Theta(n)$ برقرار باشد. درخت زیر را در نظر بگیرید که در آن برگها به مقدار $O(1)$ ختم میشوند. عددی که در کنار هر سطح درخت نوشته شده، حداکثر هزینهای است که برای تقسیم و ترکیب زیرمسأله‌ها نیاز است. این عدد برای همه سطح‌ها یکسان و حداکثر برابر cn است.



مجموع $O(n \lg n)$

طولانی ترین مسیر از ریشه به یک برگ، مسیر زیر است:

$$n \to (\frac{2}{3})n \to (\frac{2}{3})^2 n \to \cdots \to 1$$

اگر ارتفاع درخت را با h نشان دهیم، چون $(\frac{2}{3})^h n = O(1)$ است، پس ارتفاع درخت برابر با $h = O(\log_{\frac{3}{2}} n) = O(\lg n)$ خواهد بود. ما انتظار داریم که جواب رابطه بازگشتی حداکثر برابر با تعداد مرحله‌ها ضربدر هزینهی هر مرحله یا $O(cn \log_{\frac{3}{2}} n) = O(n \lg n)$ باشد. پس حدس میزنیم که کران دوطرفه مجانبی $\Theta(n \log n)$ باشد.

برای اثبات دقیق ادعای فوق باید نشان دهیم:

$$\begin{cases} I) & T(n) = O(n \log n) \\ II) & T(n) = \Omega(n \log n) \end{cases}$$

که این را می‌توان همانند مثال اول «حدس و استقرا» نشان داد.

۵. قضیه اصلی برای روابط بازگشتی

قضیه اصلی، روشی برای حل روابط بازگشتی به فرم $T(n) = aT(\frac{n}{b}) + f(n)$ است، که در آن $a \geq 0$ و $b > 0$ اعداد ثابت هستند و $f(n)$ یک تابع مثبت دارای مجانب است. در واقع می‌توان اینطور برداشت کرد که رابطه فوق، زمان اجرای یک الگوریتم با اندازه n را که به زیر مسئله‌هایی با اندازه $\frac{n}{b}$ تقسیم شده است را نشان میدهد. در این رابطه a زیر مسئله به طور بازگشتی حل میشوند و $f(n)$ نیز زمان لازم را برای تقسیم و ادغام مسئله نشان میدهد.

قضیه اصلی(Master Theorem) رابطه بازگشتی $T(n)$ بصورت $T(n) = aT(\frac{n}{b}) + f(n)$ تعریف شده است، که در آن $f(n)$ تابعی مثبت می باشد و اعداد $a \geq 0$ و $b > 0$ ثابت هستند:

۱) اگر برای $\varepsilon > 0$ ثابت $f(n)$ از $O(n^{\log_b a-\varepsilon})$ باشد؛ آنگاه $T(n)$ از $\Theta(n^{\log_b a})$ خواهدبود.

۲) اگر $f(n) = \Theta(n^{\log_b a})$ باشد؛ آنگاه $T(n) = \Theta(n^{\log_b a} \log n)$ خواهد بود.

۳) اگر برای $\varepsilon > 0$ ثابت $f(n)$ از $\Omega(n^{\log_b a+\varepsilon})$ باشد؛ آنگاه $T(n)$ از $\Theta(f(n))$ خواهدبود.

در هر کدام از این سه حالت، ما تابع $f(n)$ را با $n^{\log_b a}$ مقایسه میکنیم، و تابع بزرگتر، جواب رابطه ی بازگشتی را مشخص میکند. البته برای حالت تساوی باید ضریب $\log n$ را لحاظ کرد. برای استفاده از قضیه‌ی اصلی، کافیست مشخص کنیم که رابطه‌ی داده شده کدام یکی از ۳ حالت قضیه است، سپس جواب را محاسبه نماییم.

یک حالت خاص و کاربردی قضیه‌ی اصلی برای حل معادله بازگشتی $T(n) = aT(\frac{n}{b}) + n^d$ است.

۱) $d < \log_b a \iff a > b^d \longrightarrow T(n) = \Theta(n^{\log_b a})$

۲) $d = \log_b a \iff a = b^d \longrightarrow T(n) = \Theta(n^{\log_b a} \log n)$

۳) $d > \log_b a \iff a < b^d \longrightarrow T(n) = \Theta(n^d)$

توجه شود که ممکن است بعضی از روابط بازگشتی باشند که در شرایط قضیه اصلی صدق نکنند و برای محاسبه‌ی آنها مجبور باشیم از روشهای دیگر استفاده کنیم.

رابطه $T(n) = 2T(n/2) + n \log n$ را در نظر بگیرید. با توجه به نماد گذاری های قضیه اصلی بوضوح برای هر $\varepsilon > 0$ دلخواه، $f(n) \neq O(n^{1-\varepsilon})$ و $f(n) \neq \Theta(n)$ است. بنابراین به اشتباه ممکن است رابطه بازگشتی به عنوان حالت سوم قضیه اصلی در نظر گرفته شود، ولی دقت کنید که برای هیچ $\varepsilon > 0$ تساوی $\log n = \Omega(n^{\varepsilon})$ برقرار نیست.

برای مطالعه اثبات قضیه اصلی می‌توانید اینجا را ببینید.

۶. تمارین مروری

۱. روابط بازگشتی با پیشینه‌ی کامل زیر را حل کنید: $(T(1) = 1)$

$$\text{I.}\quad T(n) = \max_i T(i)$$

$$\text{II.}\quad T(n) = n + \sum_{i=1}^{n-1} T(i)$$

۲. ثابت کنید که به ازای هر عدد صحیح مثبت k داریم:

$$\text{I.}\quad \sum_{i=1}^{n} i^k = O(n^{k+1})$$

$$\text{II.}\quad \sum_{i=1}^{n} i^k \log_2 i = O(n^{k+1} \log n)$$

۳. رفتار تابع $S(n)$ را که در رابطه‌ی بازگشتی زیر صدق می‌کند را بیابید.

$$S(mn) \leq cm \times \log_2 m \times S(n) + O(mn), S(2) = 1$$

که c و m مقادیر ثابتی هستند (یعنی جواب باید تابعی از c، m و n باشد).

۴. رابطه‌ی بازگشتی زیر در الگوریتم‌های که مسأله به قسمت‌های نامساوی تقسیم میشود، ظاهر میشود.

$$T(n) = \sum_{i=1}^{k} a_i T(n/b_i) + cn$$

که تمامی a_iها و b_iها ثابت هستند و در شرط زیر صدق میکنند:

$$\left(1 - \sum_{i=1}^{k} a_i/b_i\right) > 0$$

سعی کنید رفتار این رابطه‌ی بازگشتی را با کمی حدس و آزمایش بیابید.

۵. دو رابطه‌ی بازگشتی زیر را حل کنید.

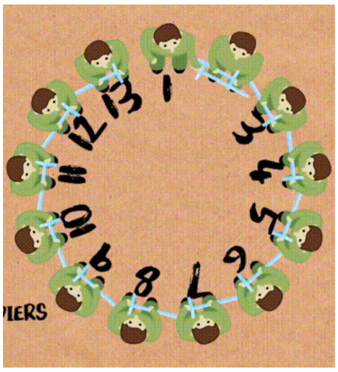
$$\text{I.}\quad T(n) = 4T(\sqrt{n}) + 1, \qquad T(2) = 1$$

$$\text{II.}\quad T(n) = 2T(\sqrt{n}) + 2n, \qquad T(2) = 1$$

۶. **(دزدان دریایی)** پنج دزد دریایی منطقی وجود دارد (به ترتیب دقیق ارشدیت ، A ، B ، C ، D و E) که ۱۰۰ سکه طلا پیدا کردند. آنها باید تصمیم بگیرند که چگونه آنها را توزیع کنند. قوانین توزیع در بین دزدان دریایی می گوید که قدیمی ترین دزد دریایی ابتدا برنامه توزیع را پیشنهاد می کند. سپس دزدان دریایی، از جمله پیشنهاد دهنده ، در مورد قبول این توزیع رای می دهند. اگر اکثریت این طرح را بپذیرند ، سکه ها پخش می شوند و بازی به پایان می رسد. در صورت رأی مساوی، پیشنهاد دهنده رأی اصلی را دارد. اگر اکثریت این طرح را رد کنند، پیشنهاد دهنده را از کشتی دزدان دریایی به دریا می ریزند و می میرد، و دزد دریایی ارشد بعدی پیشنهاد جدیدی را برای شروع مجدد ارائه می دهد. این روند تکرار می شود تا زمانی که یک طرح پذیرفته شود یا این که تنها یک دزد دریایی باقی بماند.

دزدان دریایی تصمیمات خود را بر اساس چهار عامل انجام می دهند. اول از همه، هر دزد دریایی می خواهد زنده بماند. دوم، با توجه به بقا، هر دزد دریایی می خواهد تعداد سکه های طلا را که دریافت می کند به حداکثر برساند. سوم، اگر هر نتیجه دیگری برابر باشد، هر دزد دریایی ترجیح می دهد دیگری را به دریا بیاندازد. سرانجام به نظر شما سکه ها چگونه توضیع می‌شوند؟

۷. مسئله دزدان دریایی را برای ۱۰۰ دزد و ۱۰۰۰ سکه حل کنید.



۸. (ژوزفوس) افرادی را در نظر بگیرید که دایره وار ایستاده‌اند و منتظر اعدام هستند. بعد از آنکه اولین نفر اعدام می‌شود، تعداد مشخصی از افراد رد شده و یک نفر دیگر اعدام می‌شود. سپس دوباره به همان تعداد افراد پرش شده و نفر بعد کشته می‌شود. این فرایند حذف، دور دایره(که با برداشتن افراد کشته شده کوچک و کوچکتر می‌گردد) ادامه می یابد تا زمانی که تنها یک نفر باقی می‌ماند که آزاد می‌شود. مطلوب، یافتن جایگاهی در دایره اولیه است که شما با قرار گرفتن در آنجا نجات خواهید یافت.

۷. برای جستجو

1. bread–and–butter classes of complexity
2. Inverse Ackerman's function
3. Taxicab Number
4. Akra-Bazzi method
5. Cook-Levin theorem