دانشکده‌ی علوم ریاضی

ساختمان داده‌ها — ۶ مهر ۹۲

جلسه‌ی ۵: حل روابط بازگشتی

مدرّس: دکتر شهرام خزائی — *نگارنده: آرمیتا ثابتی‌اشرف و علی‌رضا علی‌آبادیان*

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 سخت‌تر هم می‌شود. برای محاسبه‌ی زمان اجرای الگوریتم‌های تقسیم و حل، لازم است که کران مجانبی روابط بازگشتی آن‌ها رو پیدا کنیم. در این بخش تکنیک‌هایی برای پیدا کردن کران‌های روابط بازگشتی معرفی می‌کنیم و در انتها قضیه‌ی اصلی عنوان می‌شود که در پیدا کردن کران مجانبی طیف وسیعی از توابع کاربرد دارد.
در زیر نمونه‌های ساده‌ای از روابط بازگشتی به همراه جواب آن‌ها مشاهده می‌شوند:

$$T(n)=\begin{cases}1 & n=1\\ T(n-1)+1 & n\geq 2\end{cases}\qquad T(n)=n$$

$$T(n)=\begin{cases}1 & n=1\\ 2T(\frac{n}{2})+n & n\geq 2\end{cases}\qquad T(n)=n\log n+n$$

$$T(n)=\begin{cases}0 & n=2\\ T(\sqrt{n})+1 & n\geq 3\end{cases}\qquad T(n)=\log\log n$$

$$T(n)=\begin{cases}1 & n=1\\ T(\frac{n}{3})+T(\frac{2n}{3})+n & n\geq 2\end{cases}\qquad T(n)=\Theta(n\log n)$$

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.

نکته ۱ *وقتی ما مساله‌ای را به صورت مجانبی حل می‌کنیم دیگر برایمان مقدار اولیه مهم نیست، البته اگر موقع اثبات استقرایی به مقدار ثابتی جز مقدار ثابت اولیه (فرض شده) برسیم اثباتمان غلط است.*

نکته ۲ *روشی کلی برای این که حدس خوبی در مورد جواب رابطه‌ی بازگشتی بزنیم وجود ندارد، ولی اگر رابطه‌ای، شبیه به رابطه‌ای که قبلا دیده‌ایم باشد، بهتر است که حدس ما شبیه به جواب رابطه‌ی قبلی باشد.*

مثال ۱ *کران بالای مجانبی تابع $T(n)=4T(\frac{n}{4})+n$ چیست؟*
به اشتباه حدس می‌زنیم که کران بالای مجانبی $\mathcal{O}(n)$ باشد. بنابراین فرض ناصحیح استقرا به صورت زیر است:
"ضریب ثابت c وجود دارد به‌طوریکه برای n های به اندازه کافی بزرگ داریم $T(n)\leq cn$"
حال ببینیم چه مشکلی در روند اثبات گذار استقرا پیش می‌آید.

- ***پایه:*** *با توجه به این که $T(1)=\mathcal{O}(1)$ 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:

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

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

مثال ۲ 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:

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

- 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 طوری‌که برای n های به اندازه کافی بزرگ $T(n) \leq dn \log n$"
حال با دنبال کردن اثبات گذار استقرا شرایط لازم برای وجود چنین dای را تعیین می‌کنیم:

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

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 طوری‌که برای n های به اندازه کافی بزرگ $T(n) \geq dn \log n$"
حال با دنبال کردن اثبات گذار استقرا شرایط لازم برای وجود چنین dای را تعیین می‌کنیم:

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

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.

در نتیجه کران دوطرفه مجانبی تابع $T(n)$ که در رابطه $T(n) = 2T(\frac{n}{2}) + \Theta(n)$ صدق می‌کند، $\Theta(n \log n)$ است.

مثال ۳ کران بالای مجانبی تابع $T(n) = 8T(\frac{n}{2}) + \Theta(n^2)$ چیست؟
حدس می‌زنیم که کران بالای مجانبی $\mathcal{O}(n^3)$ باشد.
فرض کنید c ضریب ثابتی باشد که به ازای آن رابطه‌ی $T(n) \leq 8T(\frac{n}{2}) + cn^2$ برای n های به اندازه‌ی کافی بزرگ برقرار است. فرض استقرا را به صورت زیر در نظر می‌گیریم:
"ضریب ثابت d وجود دارد به طوری‌که برای n های به اندازه کافی بزرگ $T(n) \leq dn^3$"
با توجه به این‌که رفتار مجانبی تابع تغییر نمی‌کند، می‌توانیم برای راحتی در اثبات فرض استقرا را به صورت زیر اصلاح نماییم:
"ضرایب ثابت d و d' وجود دارند به طوری‌که برای n های به اندازه کافی بزرگ $T(n) \leq dn^3 - d'n^2$"
حال با دنبال کردن اثبات گذار استقرا شرایط لازم برای وجود چنین d و d'ای را تعیین می‌کنیم:

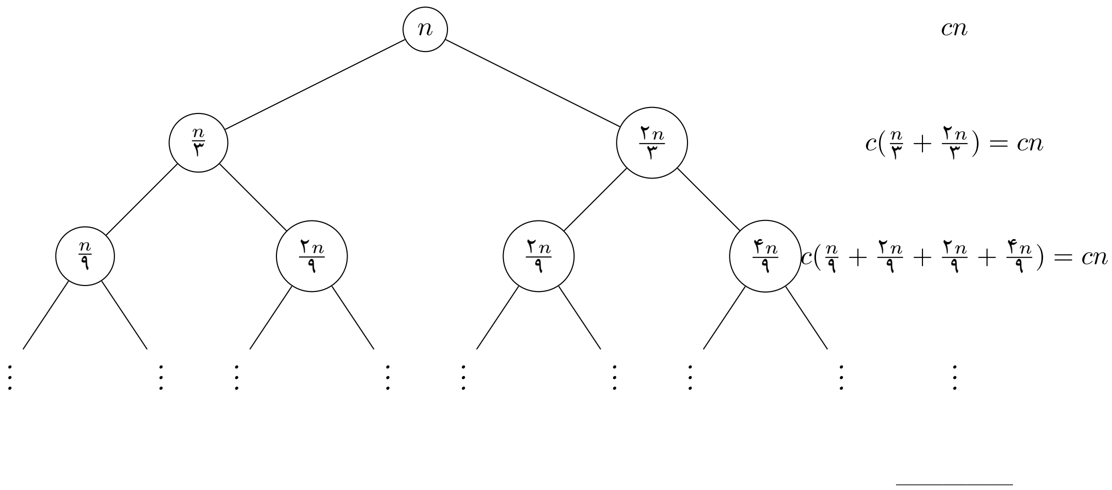
$$
\begin{aligned}
T(n) &\leq 8T(\tfrac{n}{2}) + cn^2 \\
&\leq 8(d(\tfrac{n}{2})^3 - d'(\tfrac{n}{2})^2) + cn^2 \\
&= dn^3 - d'n^2 + cn^2
\end{aligned}
$$

رابطه‌ی فوق نشان می‌دهد که اگر $c \geq d' \geq d$ انتخاب شود، فرض $T(n) \leq dn^3 - d'n^2$ صحیح است. در نتیجه کران بالای مجانبی تابع $T(n)$ که در رابطه $T(n) = 8T(\frac{n}{2}) + \Theta(n^2)$ صدق می‌کند، $\mathcal{O}(n^3)$ است.

۲ 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 امرند 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 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 ۴ *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 $T(n) \leq T(\frac{n}{3}) + T(\frac{2n}{3}) + cn$ برقرار باشد. درخت زیر را در نظر بگیرید که در آن برگ‌ها به مقدار $\mathcal{O}(1)$ 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 $O(n \lg n)$

طولانی‌ترین مسیر از ریشه به یک برگ، مسیر زیر است:

$$n \to (\tfrac{2}{3})n \to (\tfrac{2}{3})^2 n \to \ldots \to 1$$

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

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

- ابتدا (I) را اثبات می‌کنیم. فرض کنید c ضریب ثابتی باشد که به ازای آن رابطه‌ی $T(n) \leq T(\frac{n}{3}) + T(\frac{2n}{3}) + cn$ برای n های به اندازه‌ی کافی بزرگ برقرار است. فرض استقرا را به صورت زیر در نظر می‌گیریم:
”ضریب ثابت d وجود دارد به طوری‌که برای n های به اندازه کافی بزرگ $T(n) \leq dn \log n$“
حال با دنبال کردن اثبات گذار استقرا شرایط لازم برای وجود چنین dای را تعیین می‌کنیم:

$$\begin{array}{rcl} T(n) \leq dn \log n \Rightarrow T(n) & \leq & T(\frac{n}{3}) + T(\frac{2n}{3}) + cn \\ & \leq & d\frac{n}{3} \log \frac{n}{3} + d\frac{2n}{3} \log \frac{2n}{3} + cn \\ & = & d\frac{n}{3} \log n - d\frac{n}{3} log 3 + d\frac{2n}{3} \log n - d\frac{2n}{3} \log 3 + cn \\ & = & dn \log n - dn \log 3 + cn \end{array}$$

رابطه‌ی فوق نشان می‌دهد که اگر $c \leq d \log 3$ 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 $T(n) \geq T(\frac{n}{3}) + T(\frac{2n}{3}) + cn$ برای n های به اندازه‌ی کافی بزرگ برقرار است. فرض استقرا را به صورت زیر در نظر می‌گیریم:
”ضریب ثابت d وجود دارد به طوری‌که برای n های به اندازه کافی بزرگ $T(n) \geq dn \log n$“
حال با دنبال کردن اثبات گذار استقرا شرایط لازم برای وجود چنین dای را تعیین می‌کنیم:

$$\begin{array}{rcl} T(n) \geq dn \log n \Rightarrow T(n) & \geq & T(\frac{n}{3}) + T(\frac{2n}{3}) + cn \\ & \geq & d\frac{n}{3} \log \frac{n}{3} + d\frac{2n}{3} \log \frac{2n}{3} + cn \\ & = & d\frac{n}{3} \log n - d\frac{n}{3} log 3 + d\frac{2n}{3} \log n - d\frac{2n}{3} \log 3 + cn \\ & = & dn \log n - dn \log 3 + cn \end{array}$$

رابطه‌ی فوق نشان می‌دهد که اگر $c \geq d \log 3$ انتخاب شود، فرض $T(n) \geq dn \log n$ صحیح است. بنابراین (II) هم برقرار است.

در نتیجه کران دوطرفه مجانبی تابع $T(n)$ که در رابطه $T(n) = T(\frac{n}{3}) + T(\frac{2n}{3}) + \Theta(n)$ صدق می‌کند، $\Theta(n \log n)$ است.

۳ قضیه اصلی برای حل روابط بازگشتی

قضیه اصلی، روشی برای حل روابط بازگشتی به فرم $T(n) = aT(\frac{n}{b}) + f(n)$ است، که در آن $a \geq 1$ و $b > 1$ اعداد ثابت هستند و $f(n)$ یک تابع مثبت دارای مجانب است. رابطه فوق، زمان اجرای یک الگوریتم با اندازه n را که به a زیر مسئله با اندازه n/b تقسیم شده است 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 ۱ (قضیه اصلی) *رابطه بازگشتی $T(n)$ بصورت $T(n) = aT(\frac{n}{b}) + f(n)$ تعریف شده‌است، که در آن $f(n)$ تابعی مثبت می‌باشد و اعداد $a \geq ۱$ و $b > ۱$ ثابت هستند:*

- *اگر برای $\epsilon > ۰$ ثابت $f(n)$ از $\mathcal{O}(n^{\log_b a - \epsilon})$ باشد؛ آنگاه $T(n)$ از $\Theta(n^{\log_b a})$ خواهدبود.*
- *اگر $f(n) = \Theta(n^{\log_b a})$ باشد؛ آنگاه $T(n) = \Theta(n^{\log_b a} \log n)$ خواهد بود.*
- *اگر برای $\epsilon > ۰$ ثابت $f(n) = \Omega(n^{\log_b a + \epsilon})$ باشد و همچنین اگر برای n های به اندازه کافی بزرگ و $c < ۱$ ثابت، رابطه $af(n/b) < cf(n)$ برقرار باشد (که شرط انتظام[۱] نامیده می‌شود)، آنگاه $T(n) = \Theta(f(n))$ خواهد بود.*

در هر کدام از این سه حالت، ما تابع $f(n)$ را با $n^{\log_b a}$ مقایسه می‌کنیم، و تابع بزرگ‌تر، جواب رابطه‌ی 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.

نکته ۳ *ممکن است n/b یک عدد اعشاری باشد، اگر به جای $T(n/b)$، $T(\lceil \frac{n}{b} \rceil)$ و یا $T(\lfloor \frac{n}{b} \rfloor)$ بنویسیم، رفتار مجانبی رابطه بازگشتی تغییر نمی‌کند، ولی برای راحتی در رابطه‌های بازگشتی الگوریتم‌های تقسیم و حل از نوشتن کف و سقف صرف‌نظر می‌شود.*

مثال ۵ *یک حالت خاص و کاربردی قضیه‌ی اصلی برای حل معادله بازگشتی $T(n) = aT(\frac{n}{b}) + n^d$ است. در این صورت سح حالت مذکور به صورت زیر است:*

$$\begin{array}{lllll} ۱) & d < \log_b a & \Leftrightarrow a > b^d & \rightarrow & T(n) = \Theta(n^{\log_b a}) \\ ۲) & d = \log_b a & \Leftrightarrow a = b^d & \rightarrow & T(n) = \Theta(n^{\log_b a} \log n) \\ ۳) & d > \log_b a & \Leftrightarrow a < b^d & \rightarrow & T(n) = \Theta(n^d) \end{array}$$

مثال ۶ *با استفاده از قضیه‌ی اصلی، کران مجانبی توابع زیر را محاسبه کنید.*

۱. $T(n) = ۵T(\frac{n}{۲}) + \Theta(n^۲)$

$$۵ > ۲^۲ \xrightarrow{\text{حالت ۱}} T(n) = \Theta(n^{\log_۲ ۵})$$

۲. $T(n) = ۲۷T(\frac{n}{۳}) + \Theta(n^۳)$

$$۲۷ = ۳^۳ \xrightarrow{\text{حالت ۲}} T(n) = \Theta(n^۳ \log n)$$

۳. $T(n) = ۵T(\frac{n}{۲}) + \Theta(n^۳)$

$$۵ < ۲^۳ \xrightarrow{\text{حالت ۳}} T(n) = \Theta(n^۳)$$

نکته ۴ *ممکن است بعضی از روابط بازگشتی باشند که در شرایط این قضیه صدق نکنند و برای محاسبه‌ی آن‌ها مجبور باشیم از روش‌های دیگر استفاده کنیم.*

۱. *رابطه $T(n) = ۲T(n/۲) + n \log n$ را در نظر بگیرید. با توجه به نماد گذاری‌های قضیه اصلی بوضوح برای هر $\epsilon > ۰$ دلخواه، $f(n) \neq \mathcal{O}(n^{۱-\epsilon})$ 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 $\epsilon > ۰$ تساوی $\log n = \Omega(n^\epsilon)$ برقرار نیست.*

[۱] regularity condition

۲. *برای رابطه بازگشتی $T(n) = 27T(\frac{n}{3}) + \Theta(\frac{n^3}{\log n})$ نیز قضیه اصلی کاربردی ندارد. زیرا باید $n^{\log_3 27}$ را با $\Omega(\frac{n^3}{\log n})$ مقایسه کرد. از آنجایی که برای هیچ ϵ مثبتی n^{ϵ} از $\mathcal{O}(\log n)$ نیست؛ تابع $f(n) = \Theta(\frac{n^3}{\log n})$ نیز از $\mathcal{O}(n^{3-\epsilon})$ نخواهد بود.*

۳. *در مورد رابطه بازگشتی $T(n) = T(n/2) + n(2 - \cos n)$ هم نمی‌توان قضیه اصلی را به کار برد. زیرا با آنکه برای هر $1 \leq \epsilon$ تابع $f(n) = \Omega(n^{0+\epsilon})$ است؛ ولی شرط انتظام برای $f(n)$ برقرار نیست.*

۴ دنباله بازگشتی و محاسبه بازگشتی

در الگوریتم‌های بازگشتی و به خصوص هنگام محاسبه‌ی حاصل یک عبارت به صورت بازگشتی (مانند سری فیبوناچی) معمولا پیچیدگی راه‌حل مستقیم و بازگشتی باهم متفاوت است. الگوریتم‌های بازگشتی علی‌رغم اینکه گاهی راحت‌تر پیاده‌سازی می‌شوند، اغلب کندترند زیرا در آن‌ها الگوریتم به ازای یک مقدار بعضا چندبار تکرار می‌شود. مثلا در محاسبه‌ی عدد پنجم فیبوناچی به روش بازگشتی، ۳ بار عدد دوم فیبوناچی و ۲ بار عدد سوم این سری محاسبه می‌گردند ولی در راه‌حل مستقیم آن، هر اندیس این سری یک بار حساب می‌شود.

۱.۴ راه حل بازگشتی سری فیبوناچی

Algorithm 1 Algorithm: RecursiveFibonacci

```
function Fib(n)
    [assumes n ≥ 1]
    if n ≥ 3 then
        x ← Fib(n − 1)
        y ← Fib(n − 2)
        return x + y
    else
        return 1
```

پیچیدگی زمانی این روش $\Theta((\frac{1+\sqrt{5}}{2})^n)$ است، زیرا حالات تکراری در آن محاسبه می‌شوند. به همین دلیل باید کد این برنامه را به طور مستقیم پیاده‌سازی کنیم و از روش بازگشتی استفاده نکنیم.

۲.۴ راه حل مستقیم سری فیبوناچی

Algorithm 2 Algorithm: Fibonacci

```
function Fib(n)
    F[0] = F[1] = 1
    for i = 2 to n do
        F[i] = F[i − 1] + F[i − 2]
    return F[n]
```

پیچیدگی زمانی این روش $\Theta(n)$ است، زیرا برای محاسبه‌ی nامین عدد فیبوناچی، اعداد قبل از آن فقط یک‌بار محاسبه می‌شوند و کاری تکراری انجام نمی‌دهیم.