

به نام یگانه هستی بخش
دوره‌ی آموزشی المپیاد کامپیوتر
داینامیک

ما قبلاً دیده‌ایم که می‌توان یک مسئله کلی را شکست و به یک سری زیر مسئله تبدیل کرد که با حل این زیر مسئله‌ها و ادغام جواب آنها می‌توان جواب مسئله اصلی را بدست آورد. خیلی اوقات ساختار یک مسئله ما را برای شکستن آن راهنمایی می‌کند، که ما به یک سری مسئله مستقل می‌رسیم و آنها را حل می‌کنیم و آنها ما را برای حل مسئله اصلی کمک می‌کنند. ولی خیلی اوقات ما به یک سری زیر مسئله مستقل نمی‌رسیم و بلکه این زیر مسائل خیلی اوقات اشتراکات زیادی با هم دارند. در اینصورت وقتی مستقلاً این زیر مسئله‌ها را حل می‌کنیم، حالات زیادی اتفاق می‌افتند که ما در هر زیر مسئله آنها را حل می‌کنیم در حالی که یک بار حل کردن آنها کافی بوده است. در واقع در صورتی که زیر مسئله‌هایی که ما داریم مستقل از یکدیگر نباشند، با ادامه این عمل زیر مسئله‌های زیادی پیدا می‌شوند که با هم یکی هستند ولی ما همه آنها را مستقلاً یک بار حل می‌کنیم. اگر بتوانیم این جور هزینه‌ها را کم کنیم و یا اصلاً از بین ببریم، می‌توان به الگوریتم‌های بسیار بهتر و قوی‌تری رسید. ایده اصلی الگوریتم‌های داینامیک نیز همین نکته ساده می‌باشد: «پرهیز از محاسبه دوباره چیزهای یکسان». معمولاً با نگهداری جدولی از داده‌های بدست آمده که از زیر مسئله‌هایی که ایجاد کرده‌ایم آن را پر می‌کنیم، می‌توانیم به این مهم دست پیدا بکنیم.

روش داینامیک یک روش از پایین به بالا می‌باشد. ما معمولاً از آسان‌ترین و مشخص‌ترین زیر مسائل کار را شروع می‌کنیم. با ادغام کردن جواب آنها به جواب مسئله بزرگ‌تر می‌رسیم، و این کار را ادامه می‌دهیم تا آخر به جواب مسئله اصلی خواهیم رسید. روش تقسیم-حل یک روش از بالا به پایین است. وقتی یک مسئله را می‌خواهیم با این روش حل کنیم، ما فوراً به مسئله اصلی حمله می‌کنیم، سپس آن را به یک سری زیر مسئله تقسیم کرده و دوباره به این زیر مسائل حمله می‌کنیم و این کار را ادامه می‌دهیم تا مسئله حل شود.

مسئله‌ی اول: دنباله فیبوناچی $O(n)$

مقدار تابعی که به شکل زیر تعریف می شود را برای عدد n بدست آورید:

$$f(n) = \begin{cases} f(n-1) + f(n-2) & n \geq 2 \\ 1 & n = 0, 1 \end{cases}$$

اگر از تابع زیر برای محاسبه این تابع استفاده کنیم، زمان اجرای آن چقدر است؟

function F(n)
<pre>if n < 2 then return 1 else return F(n-1) + F(n-2)</pre>

مسئله‌ی دوّم: ترکیب $O(nk)$

مقدار ترکیب k از n را با استفاده از رابطه زیر بدست آورید:

$$\binom{n}{k} = \begin{cases} \binom{n-1}{k-1} + \binom{n-1}{k} & 0 < k < n \\ 1 & otherwise \end{cases}$$

اگر از تابع زیر برای محاسبه این ترکیب استفاده کنیم، زمان اجرای آن چقدر است؟

function C(n,k)
<pre>if k = 0 or k = n then return 1 else return C(n-1,k-1) + C(n-1,k)</pre>

روش دینامیک معمولاً برای حل مسائل بهینه‌سازی استفاده می شود. یعنی مسائلی که می‌خواهیم مقدار یک تابعی را کمینه یا بیشینه بکنیم. معمولاً به جوابی که مقدار بهینه را دارد، جواب بهینه گفته می شود.

ساختار یک الگوریتم دینامیک می تواند به مراحل زیر شکسته شود:

(۱) شناخت ساختار جواب بهینه

(۲) به طور بازگشتی مقدار جواب بهینه را پیدا کنیم.

(۳) مقدار جواب بهینه را به ترتیب و مدل پایین به بالا محاسبه کنیم.

(۴) ساختن جواب بهینه از اطلاعات بدست آمده.

تمرین اول: بزرگترین زیر دنباله غیر نزولی $O(n^2)$

یک الگوریتم برای پیدا کردن بزرگترین زیر دنباله غیر نزولی دنباله اعداد $X = \langle x_1, \dots, x_n \rangle$ بدهید.

تمرین دوم: نوشتن متن $O(n^2)$

فرض کنید یک متن به ما داده شده است که دارای n کلمه است که کلمه i ام آن دارای طول l_i می باشد. همچنین ما یک دفتر داریم که در آن یک سری خط قرار دارد. در هر خط از این دفتر می توان M حرف نوشت. اگر در خطی هر دو کلمه i ام و $i + 1$ ام قرار داشتند، باید بین این دو کلمه یک فاصله قرار داده شود. به این ترتیب اگر کلمات i ام تا j ام در یک خط بیایند ($i \leq j$) تعداد حروفی که در این خط هست $j - i + \sum_{k=i}^j l_k$ می باشد. تعداد حروفی که هر خط می نویسیم نباید از M بیشتر شود. هدف کمینه کردن مجموع تعداد فاصله های خالی می باشد که در انتهای خطوطی که متن را در آنها نوشته ایم قرار دارد. مثلاً اگر کلمات i ام تا j ام در یک خط بیایند ($i \leq j$) تعداد فاصله خالی که در این خط هست $M - j + i - \sum_{k=i}^j l_k$ می باشد. الگوریتمی برای پیدا کردن روشی برای نوشتن این متن پیدا کنید.

تمرین سوم: مهمانی آقای رئیس $O(n)$

رئیس یک شرکت قصد دارد یک مهمانی برپا کند. در این شرکت هر نفر دقیقاً یک رئیس دارد به غیر از خود رئیس شرکت، به این ترتیب ساختار این شرکت به صورت یک درخت ریشه دار است که ریشه آن رئیس شرکت و پدر هر راس، رئیس آن شخص است. همچنین نفر x که در این مهمانی بیاید به مقدار c_x شرکت پیشرفت می کند. طبق تحقیقاتی که انجام شده رئیس شرکت مقدار c_x ها را پیدا کرده است. همچنین دوست ندارد که یک نفر از کارکنان شرکت به همراه رئیسش در این مهمانی دعوت شوند. حال هدف او دعوت کردن تعدادی از کارکنان شرکت است که بعد از این مهمانی شرکت بیشترین پیشرفت را داشته باشد. الگوریتمی برای محاسبه افرادی که لازم است در مهمانی دعوت شوند تا شرکت بیشترین پیشرفت را داشته باشد بدست آورید.

تمرین چهارم: رودخانه و قایق ها $O(n^2)$

در رودخانه کارون n نقاط وجود دارد که در آنها می توان قایق اجاره کرد. فرض کنید این نقاط در راستای رودخانه به ترتیب از ۱ تا n شماره گذاری شده اند. همچنین فرض کنید هزینه اجاره کردن قایق از نقطه i و رفتن تا خانه j ام a_{ij} باشد. روشی ارائه دهید که با کمترین هزینه از نقطه ۱ با اجاره کردن یک سری قایق به نقطه n ام برسیم.

تمرین پنجم: رابطه $O(n^2)$

فرض کنید n شیء در اختیار داریم که می‌توانیم به ترتیبی آنها را پشت سر هم قرار داده و بین آنها علائم $<$ و $=$ را قرار دهیم. تعداد راهای انجام این کار را بدست آورید. مثلاً برای $n = 2$ ، ۳ روش و برای $n = 3$ ، ۱۳ روش وجود دارد.