طراحی و تحلیل الگوریتم

استاد:

دکتر زاهد رحمتی

تدریسیاران:

اشکان ودادی

داریوش کاظمی

ترم دوم ۱۴۰۰

جلسه چهارم

برنامه نویسی پویا (DP) – قسمت سوم

روش DP:

در روش پویا ابتدا نمونه‌های کوچکتر را حل کرده و نتایج را **ذخیره** می‌کنیم. بعدا هر زمانی به آن‌ها نیاز شد به جای دوباره حساب کردن، از داده ذخیره شده استفاده می‌کنیم.

- در DP از جدول یا آرایه استفاده میشود.
- مشابه تقسیم و حل

1- پیدا کردن رابطه بازگشتی
2- حل نمونه از مسئله به شیوه پایین به بالا با حل نمونه‌های کوچکتر (یا از پایین به بالا)

هر مسئله بهینه‌سازی را نمی‌توان با استفاده از برنامه نویسی پویا حل کرد. اصل بهینگی باید در مسئله صدق کند. گفته می‌شود اصل بهینگی در یک مسئله صدق می‌کند اگر یک حل بهینه برای نمونه‌ای مسئله، همواره حاوی حل بهینه برای همه زیرنمونه‌ها باشد.

روش پیاده سازی DP:

مرحله اول: پیدا کردن رابطه بازگشتی برای مسئله
مرحله دوم: مشخص کردن ابعاد و اندازه جدول (آرایه، لیست)
مرحله سوم: تعریف هر خانه جدول با کمک رابطه بازگشتی
مرحله چهارم: مقداردهی اولیه و شرایط مرزی
مرحله پنجم: نحوه بروزرسانی خانه‌ها
مرحله ششم: مشخص کردن جواب‌ها

سوال:

مسئله LCS را در نظر بگیرید. فرض کنید ما این اجازه را داریم که حداکثر k کاراکتر از رشته اول را به هر کاراکتر دلخواهی تغییر دهیم. با این امکان، طول بلندترین زیردنباله مشتک را بدست آورید.

پاسخ:

سوال:

مسئله LCS را در نظر بگیرید. فرض کنید ما این اجازه را داریم که حداکثر k کاراکتر از رشته اول را به هر کاراکتر دلخواهی تغییر دهیم. با این امکان، طول بلندترین زیردنباله مشتک را بدست آورید.

پاسخ:

در واقع مشابه همان الگوریتم LCS عمل خواهیم کرد:

1. کاراکتر آخر دو رشته یکسان باشد.
2. کاراکتر آخر دو رشته متفاوت باشد.

برای هر دو حالت میتوانیم یا کاراکتر اول را عوض کنیم یا نکنیم. پس حالت‌های متفاوت را می‌توانیم به صورت یک ماتریس سه بعدی $LCS[i][j][k]$ بنویسیم. که k برای تعداد کاراکترهایی که تغییر میدهیم و i و j برای طول رشته‌ها است.

سوال 1:

مسئله LCS را در نظر بگیرید. فرض کنید ما این اجازه را داریم که حداکثر k کاراکتر از رشته اول را به هر کاراکتر دلخواهی تغییر دهیم. با این امکان، طول بلندترین زیردنباله مشتک را بدست آورید.

پاسخ:

در واقع مشابه همان الگوریتم LCS عمل خواهیم کرد:

1. کاراکتر آخر دو رشته یکسان باشد.
2. کاراکتر آخر دو رشته متفاوت باشد.

برای هر دو حالت میتوانیم یا کاراکتر اول را عوض کنیم یا نکنیم. پس حالت‌های متفاوت را می‌توانیم به صورت یک ماتریس سه بعدی LCS[i][j][k] بنویسیم. که k برای تعداد کاراکترهایی که تغییر میدهیم و i و j برای طول رشته‌ها است.

$$LCS(i,j,k) = \begin{cases} 0 & i = 0, j = 0 \\ \max(LCS(i-1,j-1,k)+1, LCS(i-1,j-1,k-1)+2 & if\ i,j > 0\ and\ x_i = y_j \\ \max(LCS(i-1,j,k), LCS(i,j-1,k), LCS(i-1,j,k-1)+1, LCS(i,j-1,k-1)+1) & if\ i,j > 0\ and\ x_i \neq y_j \end{cases}$$

سوال 2:

مسئله () Cutting Rod را در نظر بگیرید. این شرط را نیز در نظر بگیرید که اگر ما چوبی به طول L را در نقطه‌های به طول k قطع کنیم، باید هزینه‌ای معادل K*(L–K) را نیز پرداخت کنیم. الگوریتمی ارائه دهید که بیشترین سود ممکن را مشخص کند.

سوال 2:

مسئله () Cutting Rod را در نظر بگیرید. این شرط را نیز در نظر بگیرید که اگر ما چوبی به طول L را در نقطه‌های به طول k قطع کنیم، باید هزینه‌های معادل (L–K)*K را نیز پرداخت کنیم. الگوریتمی ارائه دهید که بیشترین سود ممکن را مشخص کند.

پاسخ:
مشابه الگوریتم قدیم است ولی با کمی تغییر اصلاح میشود.

ماکسیموم درآمد از میله به طول j:

$$r_j = \max_{1 \le i \le j} \{P_i - i(n-i) + r_{j-1}\}, \qquad r_0 = 0$$

سوال 2:

مسئله () Cutting Rod را در نظر بگیرید. این شرط را نیز در نظر بگیرید که اگر ما چوبی به طول L را در نقطه‌های به طول k قطع کنیم، باید هزینه‌ای معادل K*(L–K) را نیز پرداخت کنیم. الگوریتمی ارائه دهید که بیشترین سود ممکن را مشخص کند.

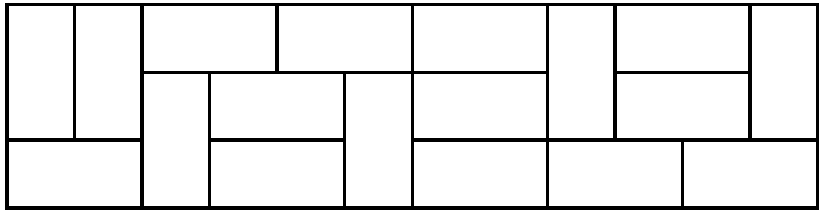
شبه کد:

```
Bottom–Up–Cut Rod(p,n):
        let r[0…n] be a new array
        r[0]=0
        for j=1 to n:
                q=infinite
                for i=1 to j:
                        q=max(q,p[i]–i*(n–1)+r[j–i])
                r[j]=q
        return r[n]
```

سوال3 – Tri Tiling:

تعداد راهای پوشاندن یک جدول 3*n را با دومینو (1*2) را بیابیید (تصویر داده شده 12*3 است).

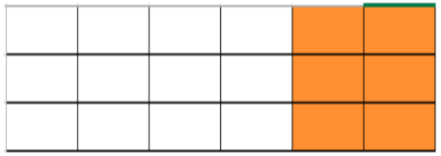
- n عدد زوج است.
- مستطیل در ابتدا خالی است.



سوال3 – Tri Tiling:

تعداد راهای پوشاندن یک جدول 3*n را با دومینو (2*1) را بیابیید (تصویر داده شده 3*12 است).

- n عدد زوج است.
- مستطیل در ابتدا خالی است.



پاسخ:

حالت1:

برای پاسخ مستطیل روبه را در نظر بگیرید. اگر دو ستون سمت راست را رنگ کنیم 2 اتفاق می‌افتد.

1. f(n–2) حالت برای سمت چپ (سفید) وجود دارد.
2. 3 حالت برای سمت نارنجی داریم.
    1. دوتا عمودی بالا، یکی افقی پایین
    2. دوتا عمودی پایین، یکی افقی بالا
    3. سه تا افقی

پس در کل 3*f(n–2) حالت داریم.

سوال3 – Tri Tiling:

تعداد راهای پوشاندن یک جدول 3*n را با دومینو (1*2) را بیابیید (تصویر داده شده 12*3 است).

- n عدد زوج است.
- مستطیل در ابتدا خالی است.

پاسخ:

حالت2:

اگر 4 ستون سمت راست را رنگ کنیم 2 اتفاق می‌افتد (وضعیت های حالت 1 را در نظر نمیگیریم).

1. f(n–4) حالت برای سمت چپ (سفید) وجود دارد.
2. 2 حالت برای سمت نارنجی داریم. (تصویر داده شده)

پس در کل 2*f(n–4) حالت داریم.

سوال3 – Tri Tiling:

تعداد راهای پوشاندن یک جدول 3*n را با دومینو (2*1) را بیابیید (تصویر داده شده 3*12 است).

- n عدد زوج است.
- مستطیل در ابتدا خالی است.

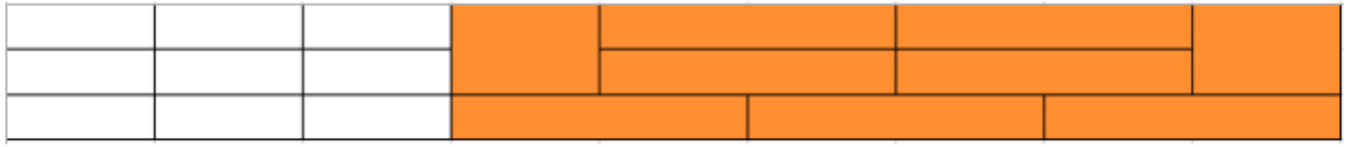
پاسخ:

حالت 3:

اگر 6 ستون سمت راست را رنگ کنیم 2 اتفاق می‌افتد (وضعیت های حالت 1 و 2 را در نظر نمیگیریم).

1. f(n–6) حالت برای سمت چپ (سفید) وجود دارد.
2. 2 حالت برای سمت نارنجی داریم. (تصویر داده شده و برعکس آن)

پس در کل 2*f(n–6) حالت داریم.



سوال3 – Tri Tiling:

تعداد راهای پوشاندن یک جدول 3*n را با دومینو (1*2) را بیابیید (تصویر داده شده 3*12 است).

- n عدد زوج است.
- مستطیل در ابتدا خالی است.

پاسخ:

پس در نهایت می‌توان گفت (f(0)=1):

$$f(n)=3f(n-2) + 2f(n-4) + 2f(n-6) + \dots + 2f(n-(n-2)) + 2$$

و در نهایت داریم:

$$f(n)=3f(n-2) + 2f(n-4) + 2f(n-6) + \dots + 2f(2) + 2f(0)$$

حال اگر به جای n مقدار n–2 بگذاریم:

$$f(n-2)=3f(n-4) + 2f(n-6) + \dots + 2f(2) + 2f(0)$$

حال اگر این دو گزاره را از هم کم کنیم داریم:

$$\mathbf{f(n) = 4f(n-2)-f(n-4), \text{ where } f(0)=1, f(2)=3}$$

سوال3 – Tri Tiling:

تعداد راهای پوشاندن یک جدول 3*n را با دومینو (1*2) را بیابیید (تصویر داده شده 3*12 است).

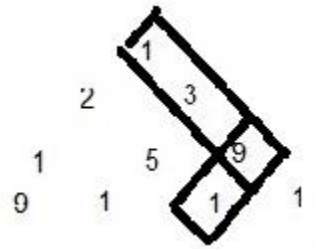
- n عدد زوج است.
- مستطیل در ابتدا خالی است.

کد:

```
int dp[31], n;
memset(dp, 0, sizeof(dp));
dp[0]=1; dp[2]=3;
while(true){
        cin>>n;
        if(n==–1) break;
        if(n%2==1)  cout<<0<<endl;
        else if(dp[n]!=0) cout<<dp[n]<<endl;
         else{
                for(int i=4; i<=n; i++){
                        dp[i] = 4*dp[i–2] – dp[i–4];
                }
                 cout<<dp[n]<<endl;
        }
}
```

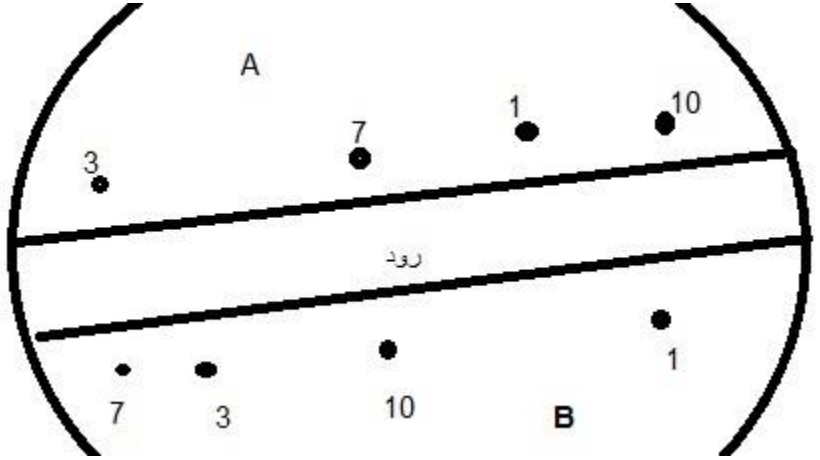
سوالات بیشتر:

1. n را از ورودی بگیر و تعداد راهای نوشتن آن با اعداد 1 و 3 و4 را بدست آورید. مثلا عدد 5 را می توان به 5 روش نوشت.
2. دنباله ای داریم از nعدد صحیح (مثبت و منفی) بزرگترین زیر آرایه متوالی با بیشترین مجموع را بیابید.
3. دنباله ای داریم از nعدد صحیح (مثبت و منفی) طول بزگترین زیر دنباله ی صعودی را بیابیید.
4. فردی دارای nنوع سکه است و می خواهد اسکناس kتومانی خود را با این سکه ها خرد کند. کمترین تعداد سکه های لازم را بیابیید. ابتدا nرا دریافت کرده و سپس nنوع سکه را دریافت کرده و kرا دریافت کنید.
5. مثلثی مانند شکل زیر داریم که دارای nلایه است می خواهیم مسیری از قله ی مثلث تا پایین انتخاب کنیم که مجموع اعداد آن بیشینه شود آن عدد را چاپ کنید.(شکل) هر عدد می تواند به دو عدد مجاور خود برود.ابتدا nرا دریافت و مثلث را دریافت کنید.



سوالات بیشتر:

6. مجموعه ای داریم از nعضو تعداد زیر مجموعه های از آن که مجموع اعضای آن برابر با cباشدرا بیابیید.
7. در یک کشور که دارای 2n شهر است و دارای یک رود می باشد متاسفانه این رود از وسط کشور رد شده است به گونه ای که n شهر در یک طرف و n شهر دیگر در طرف دیگر قرارگرفته است به همین دلیل رئیس جمهور آن کشور تصمیم گرفته که بین مجموعه ی A , B از شهرها پل احداث کنیم . هر شهر فقط می تواند با شهری پل داشته باشد که با آن هم جمعیت باشد . هر شهر می تواند با شهر دیگر پل داشته اگر با پلی دیگر برخورد نکند. حداکثر این کشور چند پل می تواند داشته باشد.ابتدا n را دریافت کن سپس دو دنباله ی nعضوی از جمعیت شهر های Aو Bدریافت کن که جمعیت شهرهای A وB را به ترتیب در آن دو دنباله دارد.نکته : میدانیم هر شهر در پایین با یک شهر با بالا هم جمعیت است.حداکثر چند پل می تواند احداث شود.



سوالات بیشتر:

8. فرض کنید جدولی 100*100 داریم که کپل در ابتدا در خانه ی پایین و سمت چپ یعنی خانه ی 1 و 1 قرار دارد و میخواهد به خانه ی سمت راست و بالا یعنی 100 و 100 برود.در هر مرحله اگه در خانه ی x , y قرار داشته باشد چنانکه x+y مضرب 5 بوده به یکی از خانه های x+2 , y or x , y+2 می رود. در غیر این صورت به یکی از خانه های x+1 , y+1 or x , y+1 خواهد رفت.
9. یک آرایه n عضوی از اعداد صحیح را در نظر بگیرید. بلندترین زیردنباله اکیدا صعودی اعداد را مشخص کنید.
10. الگوریتمی با پیچیدگی زمانی $O(n^2)$ بنویسید که یک رشته را به عنوان ورودی بگیرد و بزرگترین زیر رشته پالیندروم آن را برگردانند.
11. الگوریتمی با پیچیدگی زمانی $O(n^2)$ ارائه دهید که یک رشته بگیرد و بزرگ ترین زیر رشته که تکرار شده است و هیچ حرف مشترکی بین تکرار ها باهم مشترک نباشد، برگرداند.
12. با استفاده از روش برنامه نویسی پویا، الگوریتمی ارائه دهید که یک رشته (Sو مجموعه‌ای از کلمات (Dرا از کاربر بگیرد و تعیین کنید که آیا Sرا می توان به زیر رشته های غیر خالی یک یا چند کلمه Dتقسیم کرد یا خیر. (لغات لازم نیست منحصر به فرد باشند)
13. یک عبارت منطقی به شما داده شده است و الگوریتمی با برنامه نویسی پویا ارائه دهید که تعداد حالت‌هایی که بتوان این عبارت را پرانتزبندی کرد به طوری‌که خروجی، عبارت همیشه درست باشد.

سوالات بیشتر:

.14

یک برج پایدار با ارتفاع nبرجی است که دقیقاً از nکاشی واحد ارتفاع تشکیل شده است که به صورت عمودی روی هم چیده شده اند، به طوری که کاشی بزرگتری روی کاشی کوچکتر قرار نمی گیرد، یعنی می توانیم کاشی را با اندازه کوچکتر یا مساوی آخرین کاشی قرار دهیم. روی پشته قرار می گیرد. مشابه تصویر زیر:

ما بی نهایت کاشی در اندازه های 1، 2 ،3،...، mداریم. کار محاسبه تعداد برج‌های ثابت مختلف با ارتفاع nاست که می‌توان از این کاشی‌ها ساخت، با این محدودیت که می‌توانید حداکثر kکاشی در هر اندازه در برج استفاده کنید. بنابراین ما باید تعداد کل پشته های کاشی مختلف را با استفاده از حداکثر kکاشی در هر اندازه پیدا کنیم.

الگوریتمی در زمان O(n*m)ارائه دهید که تعداد حالت‌های مختلف ساختن برج با اعداد داده شده (n,k,m) را به ما بدهد.

سوالات بیشتر (پاسخ):

راهنمای سوال 6:

$$dp[i][x]=dp[i-1][x]+dp[i-1][x-ci]$$

که dp[i][x]تعداد زیرمجموعه هایی از iعضو اوله که مجموع xدارند.

راهنمای سوال 7: مثل LCSعمل میکند O(N2)

راهنمای سوال 10: Longest Palindromic Substring using Dynamic Programming (opengenus.org)

راهنمای سوال 11: Longest repeating and non overlapping substring in a string (opengenus.org)

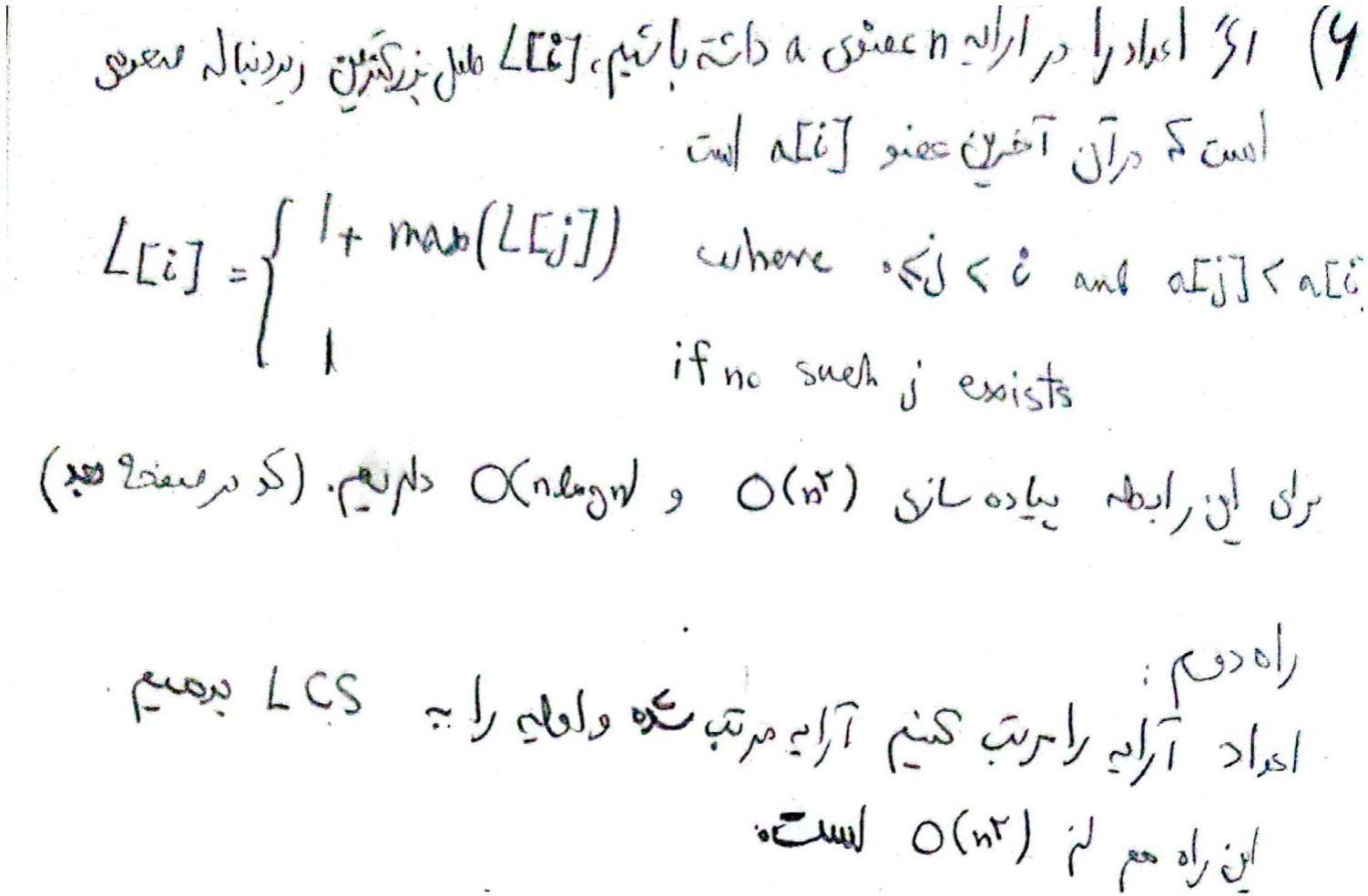
راهنمای سوال 12: Word Break Problem (opengenus.org)

راهنمای سوال 13: Boolean Parenthesization Problem (opengenus.org)

راهنمای سوال 14: Tile Stacking Problem (opengenus.org)

سوالات بیشتر (پاسخ):

راهنمای سوال 9:



خسته نباشید!

داریوش کاظمی - اشکان ودادی