تمرین سری سوم درس طراحی الگوریتم‌ها

مدرس: علی شریفی زارچی
گروه طرح سؤال: آقایان مهدی صفرنژاد بروجنی، علیرضا رضایی، اشکان نوروزی فرد، مهران خدابنده و خانم ها مریم علی اکبرپور و نرگس نوروزی

۳ آبان ۱۳۹۰

۱

سؤال

در یک کارخانه چوب بری برای اینکه یک تکه چوب را در یک مرحله به k تکه برش بزنند c_k واحد هزینه گرفته می شود. می خواهیم یک تکه چوب را با کمترین هزینه دقیقا n قسمت کنیم. الگوریتمی کارا ارائه کنید که اعداد n و بردار $\vec{c} = (c_1, c_2, ..., c_n)$ را از ورودی دریافت کند و کمترین هزینه را به دست آورد.

پاسخ

$d[i]$ را کمترین مقدار پول مورد نیاز برای برش یک قطعه چوب به k قسمت در نظر می گیریم. و از $O(n)$ آن را به روز رسانی می کنیم.
مقدار دهی اولیه : $d[0] = 0$ ، $d[1] = 0$
حال می خواهیم $d[i]$ را محاسبه کنیم. ابتدا$d[i]$ را ∞ قرار می دهیم و سپس طبق رابطه ی زیر به روز رسانی

می کنیم. طبق $psudo\ code$ زیر هر مرحله یک تکه چوب را برداشته (تکه های چوب یکسان هستند) و به j قسمت تقسیم می کنیم:

```
for(j: 1 to i-1) do
    d[i]=min(d[i],d[i-j]+c[j])
end
```

جواب برابر $d[n]$ خواهد بود.

۲

سؤال

در ماتریس $A_{m,n}$ از اعداد طبیعی، می خواهیم از گوشه سمت چپ بالا به گوشه سمت راست پایین با حرکات به راست و پایین طوری حرکت کنیم که حاصلضرب اعداد روی مسیر کمترین تعداد صفر را در سمت راست داشته باشد. الگوریتمی کارا ارائه کنید که مسیر بهینه را به دست آورد و آن را تحلیل کنید.

پاسخ

برای اینکه کمترین تعداد صفر را داشته باشیم کافیست که کمترین ۲ یا کمترین ۵ را داشته باشیم . هر دو را جدا با $dynamic\ programming$ به دست می آوریم و مینیمم می گیریم.
فرض کنید که هدف این باشد که حاصل‌ضرب اعداد روی مسیر کم‌ترین تعداد عامل ۲ رو داشته باشد. این زیر مسئله به سادگی با استفاده از $dynamic\ programming$ قابل حل است:

```
d2[0][0] = number_of_2_factors (a[0][0]);
for i in range (1 to n-1)
    d2[0][i] = d2[0][i-1] + number_of_2_factors(a[0][i]);
for j in range (1 to n-1)
    d2[j][0] = d2[j-1][0] + number_of_2_factors(a[j][0]);
for i in range (1 to n-1)
    for j in range (1 to n-1)
      d2[i][j] = min(d[i-1][j], d[i][j-1]) + number_of_2_factors (a[i][j]);
```

این زیر مسئله، با حافظه‌ی $O(n^۲)$ و در زمان $O(n^۲)$ حل می‌شود.
به طریق مشابه می‌توان حداقل تعداد عامل ۵ در مسیرهای مختلفی که برای رسیدن به خانه‌ی نهایی وجود دارد را به دست آورد.
در نهایت، پاسخ مسئله برابر است با $min(d۲(m-۱,n-۱), d۵(m-۱,n-۱))$.

۳

سؤال

یک کارخانه دارای یک سیستم بسته بندی کالا به شرح زیر است: n کالا که هر یک وزنی معادل w_i دارد به ترتیبی از پیش تعیین شده وارد مرکز بسته بندی می شود. در مرکز بسته بندی دو بسته باز قرار دارد که هر یک حداکثر P واحد وزن را در خود جای می دهد. سیستم ما در برابر کالایی که وارد می شود این عکس العمل را می تواند نشان دهد:
۱. آن را در یکی از بسته های فعلی قرار دهد.
۲. یکی از بسته ها را ببند و کنار بگذارد و یک بسته خالی مشابه را به جایش قرار دهد و کالا را در آن قرار دهد.
درانتهای کار بسته های غیر خالی فعلی را بسته و کنار می گذاریم. هدف ما کم کردن تعداد بسته هایی است که استفاده کرده ایم.
الف. الگوریتمی با زمان $O(nP^۲)$ ارائه دهید.
ب. زمان الگوریتم خود را به $O(nP)$ کاهش دهید.

پاسخ

$d[i][x][y]$ را تعریف می‌کنیم: کم‌ترین تعداد بسته‌ی لازم برای بسته‌بندی i کالای اول با فرض این‌که ظرفیت باقی‌مانده از جعبه‌ی اول برابر x و ظرفیت باقی‌مانده از دومین بسته برابر y باشد:

```
initializing:
for x in range (0 .. P)
    for y in range (0 .. P)
        d[0][x][y] = 0;

for i in range (1 .. n)
    for x in range ( 0 .. P )
        for y in range ( 0 .. P )
            if (x >= w[i])
                first_choice = d[i-1][x - w[i]][y];
            else
                first_choice = d[i-1][P - w[i]][y] + 1;
            if ( y >= w[i] )
                second_choice = d[i-1][x][y - w[i]];
            else
                second_choice = d[i-1][x][P - w[i]] + 1;
            d[i][x][y] = min (first_choice, second_choice);
```

حافظه‌ی مصرفی این الگوریتم $O(nP^۲)$ و زمان اجرا نیز $O(nP^۲)$ است.
همچنین می توان زمان و حافظه این الگوریتم را به این صورت کاهش داد که اندازه یکی از جعبه ها را به عنوان پارامتر x در نظر گرفت و $y[i][x]$ بیشترین فضای خالی از جعبه دیگر در نظر گرفت به صورتی که i

کالای اول را با استفاده از کمترین تعداد جعبه بسته بندی کرده باشیم و دو جعبه باز داریم که یکی x واحد خالی دارد و ماکسیمم جای خالی جعبه دیگر همان $y[i][x]$ می باشد.

۴

سؤال

فرض کنید مجموعه (نامتناهی) S، مجموعه ی تمام رشته های دودویی می باشد که تعداد ۱های آن ها کمتر یا مساوی L باشد. برای اعضای این مجموعه ترتیبی به این صورت تعریف می کنیم: برای دو عضو a و b در صورت تفاوت طول، رشته با طول کوچکتر، کوچکتر است و در صورت تساوی طول، رشته ای که به صورت الفبایی کوچکتر است، کوچکتر می باشد.
الگوریتمی کارآمد ارائه دهید که با گرفتن عدد k از ورودی، k اُمین عضو این مجموعه را خروجی بدهد.

پاسخ

لم: طول رشته‌ی مورد نظر حداکثر برابر $L + \log(k) + ۱$ می‌شود.
$d[i][j]$ را تعریف می‌کنیم: تعداد رشته‌های به طول i که حداکثر j تا یک دارند:

```
for j in range (0 ... L)
    d[0][j] = 1;

for i in range (1 ... L+log(k)+1)
    d[i][0] = 1
    for j in range (1 … L)
        d[i][j] = d[i-1][j] + d[i-1][j-1];
```

پس به دست اوردن مقادیر $d[i][j]$ ها این گونه عمل می کنیم . در ابتدا عدد صفر را در نظر می گیریم و از $i = ۱$ شروع می کنیم و در هر مرحله به عددی که داریم مقدار $d[i][l]$ اضافه می کنیم و i را در هر مرحله یک واحد اضافه می کنیم تا زمانی که این مقدار برای اولین بار از k بزرگ تر شود مثلا در $i = s$. پس عدد مورد نظر s رقمی است پس اگر جمع این اعداد در مرحله $i = s - ۱$ برابر sum باشد کافیست عدد $k - sum$ را در بسط دودویی در نظر بگیریم . و آن را تا s بیت در صورت لزوم گسترش بدهیم . در این صورت عدد بدست آمده جواب است . از لحاظ مرتبه زمانی نیز به وضوح در بدترین حالت $O(L \times (L + \log k + ۱))$ برای پر کردن جدول هزینه می کنیم.

۵

سؤال

N معجون داریم که می خواهیم آنها را با یکدیگر ترکیب کنیم. هر معجون عددی دارد که بین ۱ تا n است. در هر مرحله می توانیم دو معجون را با هم ترکیب کنیم و عدد معجون جدید برابر باقیمانده جمع عدد دو معجون قبل بر n خواهد شد. همچنین با ترکیب هر دو معجون مقداری دود متناسب با ضرب عدد دو معجون

تولید می شود. الگوریتمی کارا ارائه کنید که راهی را نشان دهید که همه معجونها را با یکدیگر ترکیب کنیم و کمترین دود حاصل شود.

پاسخ

این سوال را با دینامیکی نمایی حل می‌کنیم، به این شکل که یک جدول به سایز ۲^n میگیریم، و خانه‌های آن را به شکل زیر تعریف می‌کنیم.
$d[i]$ = عدد i را به صورت باینری می‌نویسیم، $d[i]$ برابر مینیمم مقدار دود تولید شده از ترکیب بطری‌ها با عددی که بیت‌هایی آن یک هست، خواهد بود.
حال پر کردن خانه‌های جدول به صورت زیر خواهد بود، به ازای هر ۲ بطری، آن دو را مخلوط کرده و معجون جدید بدست می‌آوریم، رقم متناظر با اون ۲ معجون را ۰ می‌کنیم، و رقم متناظر با معجون جدید را ۱ می‌کنیم، دوده تولید شده برای این حالت برابر دود متناظر با دینامیک زیر و دود آن ۲ معجون خواهد بود.
پس الگوریتمی از مرتبه نمایی $O(n^۲ \times ۲^n)$ دادیم.

۶

سؤال

n سرور کامپیوتری $S_۱, S_۲, ..., S_n$ داریم و می‌خواهیم فایلی را در تعدادی از این سرورها ذخیره کنیم و در اختیار کاربرآن قرار دهیم. هزینه‌ی قراردهی این فایل در سرور iام برابر با c_i است که عددی طبیعی است. حال اگر کاربری درخواست دریافت این فایل را از سرور iام بکند و فایل در این سرور وجود نداشته باشد، سرورهای $S_{i+۱}, S_{i+۲}, ..., S_n$ به ترتیب جستجو می‌شوند تا فایل مورد نظر را در یکی از این سرورها که آن را S_j می‌نامیم یافت شوند. هزینه‌ی دسترسی به سرور S_i را برابر با مقدار $j - i$ تعریف می‌کنیم. (توجه کنید که سرورهای $S_۱, S_۲, ..., S_{i-۱}$ جستجو نمی‌شوند و یک کپی از فایل در سرور S_n وجود دارد تا تمامی جستجوها پایان پذیر باشند). الگوریتمی کارآمد ارایه دهید که مشخص کند فایل را در کدام سرورها ذخیره کنیم تا مجموع هزینه‌های قراردهی و دسترسی کمینه شود.

پاسخ

یک آرایه $n \times n$ در نظر می‌گیریم. به طوری که $a[i][j]$ (برای $i \leq j$) نشان‌دهنده‌ی کمترین هزینه برای سرورهای $S_{i+۱}, S_{i+۲}, ..., S_n$ است و اولین سروری (سرور با اندیس کوچکتر) که فایل در آن ذخیره شده است نیز S_j است. واضح است که مقدار $a[n][n]$ باید برابر با C_n باشد. حال جدول را از سطر آخر به سطر اول می‌پیماییم و خانه‌های آرایه را به صورت زیر مقدار می‌دهیم:

```
for (k = j+1 to n)
    a[i-1][j] = min ( a[i][k] + (k - (i-1)) + c_k , a[i][j])
```

واضح است که جواب برابر با مقدار کمینه‌ی سطر اول است. حال اگر هنگام به روز رسانی $a[i][j]$، $parent$های مورد نظر گرفته شود به راحتی سرورهای انتخابی به دست می‌آید.

۷

سؤال

تعدادی از بچه‌های دانشکده‌ی فیزیک دور هم جمع شده‌اند تا در مورد بارش شهابی که امشب قرار است رصد کنند، برنامه‌ریزی کنند. ما اطلاعات زیر را در مورد این بارش شهابی داریم: n بارش به فاصله یک ثانیه از هم رخ می‌دهند. پس می‌توان فرض کرد شهاب iام در ثانیه iام می بارد.
دانشکده‌ی فیزیک تلسکوپ بزرگی را راه‌اندازی کرده است تا در رصدها مورد استفاده قرار بگیرد. این تلسکوپ محدودیت هایی دارد و فقط به راست و چپ از ۰ تا ۳۵۹ درجه می چرخد (می تواند دور کامل بزند) ولی نمی تواند بالا و پایین برود. رخداد jام در درجه d_j رخ می‌دهد که d_j عددی صحیح است. تلسکوپ ما نیز در ثانیه ۰ام روی درجه‌ی ۰ که همان ستاره قطبی هست، تنظیم شده است. رخداد mام مهمترین رخداد است و حتما باید رصد شود.
به دلیل اینکه این تلسکوپ ابزار پیچیده ای هست، تنها می‌تواند با سرعت یک درجه بر ثانیه حرکت کند. به همین دلیل گروه رصد انتظار ندارند که بتوانند تمامی شهاب ها را رصد کنند. بنابر این می‌خواهند حداکثر تعداد ممکن را (با توجه به محدودیت سرعت تلسکوپ و اهمیت رخداد mام) ببینند. شما باید زیر مجموعه‌ی قابل رویت از شهاب ها را بیابید که شهاب mام شامل آن باشد و بیشترین تعداد اعضا را داشته باشد. مثال: اگر مختصات رخدادها به صورت زیر باشد، مجموعه جواب برابر با $\{۱, ۳, ۶, ۹\}$ خواهد بود.

رخدادها	۱	۲	۳	۴	۵	۶	۷	۸	۹
مختصات	۱	-۴	-۱	۴	۵	-۴	۶	۷	-۲

الگوریتمی از کارآمد برای این پیدا کردن مجموعه‌ی جواب بیابید.

پاسخ

ابتدا یک آرایه ۱ در n در نظر می‌گیریم و آن را a منامیم. مقدار $a[i]$ را برابر با بیشترین تعداد رخداد قابل رویت در نظر می‌گیریم به طوری که رخداد iام حتما دیده شده باشد. واضح است که $a[۰] = ۰$ مقدار خانه‌ی iام از رابطه‌ی زیر به دست می‌آید:

$$a[i] = ۱ + max\{a[j] | j < i \& d_i - d_j < i - j\}$$

جواب مسئله در $a[n]$ وجود دارد. اگر برای هر خانه از آرایه اینکه این خانه از کدام خانه‌ی قبلی مقدار دهی شده است را نگهداری کنیم، می‌توانیم کل اعضای مجموعه‌ی جواب را نیز به دست آوریم.

۸

سؤال

الف. DFT بردار $\vec{a} = (۰, ۱, ۲, ۲, ۱, ۲, ۰, ۱)$ را محاسبه کنید.
ب. دو چند جمله ای $A(x) = ۷x^۳ - x^۲ + x - ۱۰$ و $B(x) = ۸x^۳ - ۶x + ۳$ را با استفاده از روش سریع ضرب مبتنی بر FFT در هم ضرب کنید.

پاسخ

الف.

$\hat{a_j} = \sum_{k=۰}^{N-۱} a_k.\omega_n^{jk}$ $[\omega_n = \sqrt[n]{۱}_{|n=۸} = e^{\frac{i\pi}{۴}}]$
$\vec{a} = (۰, ۱, ۲, ۲, ۱, ۲, ۰, ۱)$
$\hat{\vec{a}} : \hat{a_۰} = a_۰ + a_۱ + ... + a_۷ = ۹$
$\hat{a_j} = ۲\omega_n^۲ + \omega_n^۳ + \omega_n^۴ + \omega_n^۴$
$\hat{a_۱} = ۱ + \sqrt{۲} + ۲i$
$\hat{a_۲} = -۱$
$\hat{a_۳} = ۲\sqrt{۲} - ۱ - ۲i$
$\hat{a_۴} = ۱$
$\hat{a_۵} = ۲\sqrt{۲} - ۱ + ۲i$
$\hat{a_۶} = -۱$
$\hat{a_۷} = -۲\sqrt{۲} - ۱ - ۲i$
$\hat{a} = (۹, ۱ + \sqrt{۲} + ۲i, -۱, ۲\sqrt{۲} - ۱ - ۲i, ۱, ۲\sqrt{۲} - ۱ + ۲i, -۱, -۲\sqrt{۲} - ۱ - ۲i)$

ب. چون حاصل‌ضرب یک چندجمله‌ای درجه‌ی ۶ خواهد بود، ریشه‌های هشتم واحد را در نظر می‌گیریم و تلاش می‌کنیم مقدار چندجمله‌ای را در آن نقاط به دست آوریم.
یادآوری: ریشه‌های n ام واحد برابرند با $\omega_{i,n} = e^{\frac{۲i\pi}{n}}$
چندجمله‌ای $A(x) = ۷x^۳ - x^۲ + x - ۱۰$ را در نظر بگیرید. ابتدا مقدار دو چندجمله‌ای $A_۱(x) = ۷x + ۱$ و $A_۲(x) = -x - ۱۰$ را در نقاطی که ریشه‌ی چهارم واحد هستند به دست می‌آوریم تا با استفاده از رابطه‌ی $A(x) = xA_۱(x^۲) + A_۲(x^۲)$، مقدار چندجمله‌ای A را در نقاطی که ریشه‌ی هشتم واحد هستند، بیابیم.
دقت کنید که:
$A(\omega_{i,۸}) = \omega_{i,۸}A_۱(\omega_{i \bmod ۴,۸}) + A_۲(\omega_{i \bmod ۴,۸})$
به طریق مشابه مقدار B را نیز به ازای ریشه‌های هشتم واحد به دست می آوریم. به این ترتیب مقدار $A(x)B(x)$ به ازای ریشه‌های هشتم واحد به دست می آید.
سپس با استفاده از ماتریسی که در کلاس مطرح شده، می‌توان ضرایب چندجمله‌ای مورد نظر را به دست آورد.