باسمه تعالی

مریم وطنی

شهریور ۹۹

طراحی الگوریتم

استاد: دکتر علی رضوی

۱- تابع زمانی زیر مربوط به کدام مسئله بازگشتی می‌باشد؟

$$T(n)=\begin{cases} d & \text{if } n=1 \\ 2T(n-1)+C & \text{if } n>1 \end{cases}$$

① فاکتوریل ② دنباله فیبوناچی ③ مرتب سازی ادغامی **④ برج هانوی**

پاسخ:

این تابع زمانی مربوط به برج هانوی است.

تابع بازگشتی زیر را برای مسئله برج هانوی می‌نویسیم:

```
Void Tower (int n , peg A , peg B , peg C)
{
        // دیسک های محور A به محور B منتقل می‌شود
        if (n==1)
                Move top disk on A to C;
        else
        {
                Tower (n-1 , A , B , C);
                Move top disk on A to C;
                Tower (n-1 , B , A , C);
        }
}
```

در تابع بالا زمانی که $n=1$ باشد، در این صورت زمان اجرا الگوریتم برابر $O(1)$ خواهد بود. در صورتی که $n>1$ باشد، در این صورت زمان اجرا برابر $2T(n-1)+O(1)$ خواهد بود. بنابراین:

$$T(n)=\begin{cases} d & \text{if } n=1 \\ 2T(n-1)+C & \text{if } n>1 \end{cases}$$

بررسی سایر گزینه ها:

گزینه ①:

$$T(n)=\begin{cases} d & \text{if } n=0 \\ T(n-1)+C & \text{if } n>0 \end{cases}$$

گزینه ②:

$$Fib(n)=\begin{cases} 0 & \text{if } n=1 \\ 1 & \text{if } n=2 \\ Fib(n-1)+Fib(n-2) & \text{if } n>2 \end{cases}$$

گزینه ③:

$$T(n)=\begin{cases} d & \text{if } n=1 \\ 2T\left(n/2\right)+Cn & \text{if } n>d \end{cases}$$

۳- عملکرد تابع زیر چیست؟

```
int F (int a, int b)
{
	if (b==0)
		return (1);
	else
		return (a*F(a,b-1));
}
```

① a^b ② b^a ③ $a \times b$ ④ $a + b$

پاسخ:

می‌توان با عددگذاری عملکرد تابع را یافت.

مثلاً با فرض اینکه $a = 4$ و $b = 3$ باشد، داریم:

$$\begin{aligned} ? \quad &= F(4,3) \\ &= 4 \times F(4,2) \\ &= 4 \times 4 \times F(4,1) \\ &= 4 \times 4 \times 4 \times F(4,0) \\ \xrightarrow{b=0} &= 4 \times 4 \times 4 \times 1 \\ \Rightarrow &= 4^3 = a^b \end{aligned}$$

در نتیجه گزینه ① صحیح است.

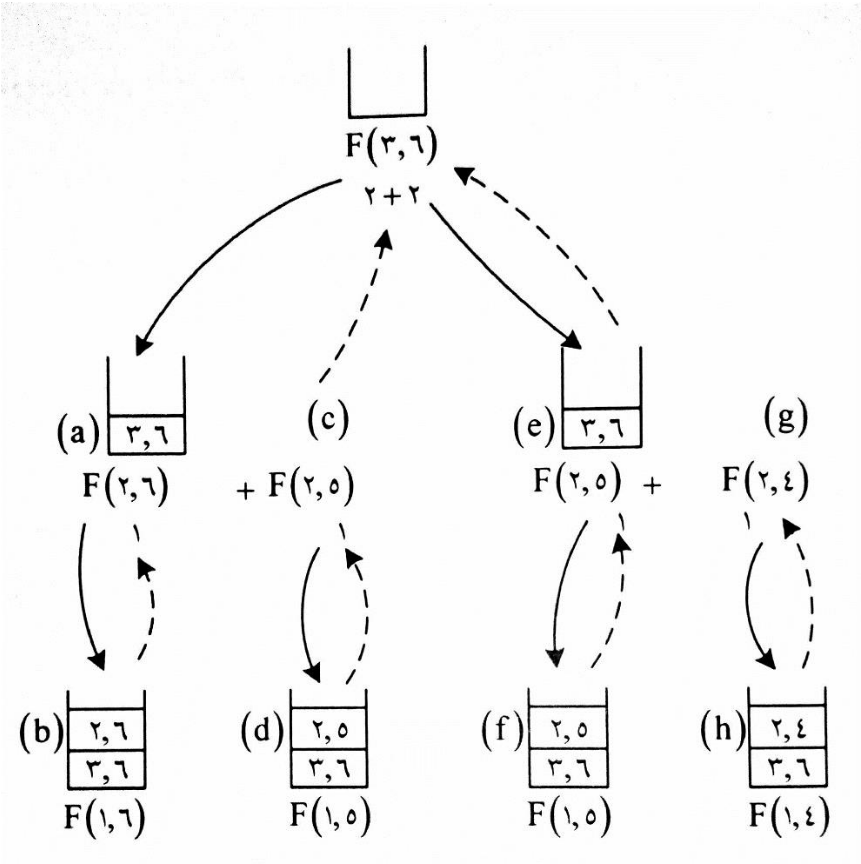
۵- **خروجی تابع زیر به ازای** F(3,6) **چیست؟**

```
int F (int m, int n)
{
        if ((m==1) | | (n==0) | | (m==n))
                return (1);
        else
                return (F(m-1, n) + F(m-1,n-1));
```

۱) 3 ۲) 4 ۳) 5 ۴) 6

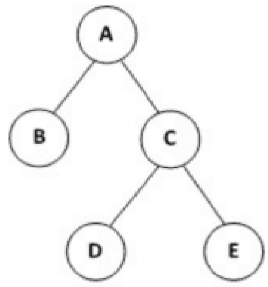
پاسخ:

مراحل اجرای الگوریتم بالا را به ازای مقادیر داده شده، در شکل زیر نمایش می‌دهیم:



شکل بالا مراحل محاسبه مقدار تابع بازگشتی را در دو مرحله فراخوانی و بازگشت نشان می‌دهد و در نهایت مقدار 4 را به عنوان خروجی نمایش می‌دهد.

۷- خروجی تابع بازگشتی زیر برای درخت دودویی شکل زیر چیست؟

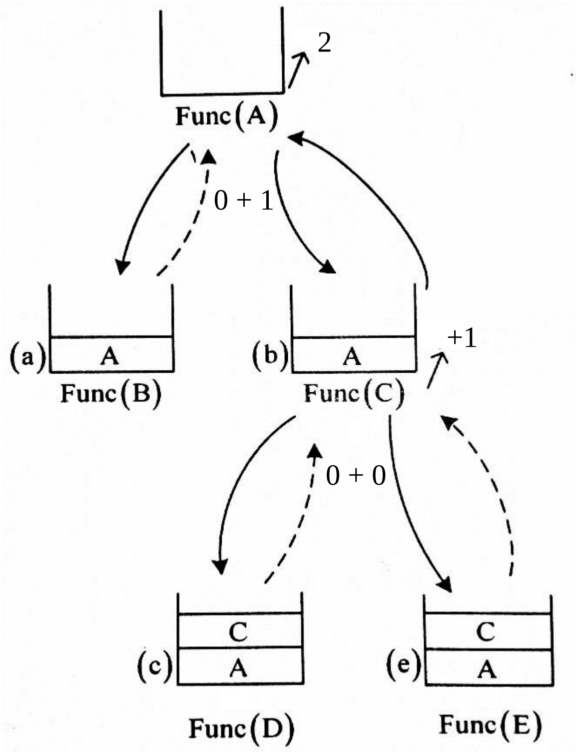



```
int func (Node *tree)
{
    if(tree==Null) return 0;
    else
    {
        if (tree → left==Null && tree → right==Null)    return 0;
        else    return (func (tree→left) + func (tree→right) +1);
    }
```

① 2 ② 3 ③ 4 ④ 5

پاسخ:

مراحل اجرایی الگوریتم بالا را به ازای درخت داده شده در شکل زیر نمایش می‌دهیم.



بنابراین گزینه ① صحیح است.

۹- مرتبه زمانی تابع بازگشتی زیر چیست؟

$$T(n)=3T\left(\frac{n}{4}\right)+n$$

① $\theta(n)$ ② $\theta(n^2)$ ③ $\theta\left(n^{\log_3^4}\right)$ ④ $\theta\left(n^{\log_4^3}\right)$

پاسخ:

طرف راست رابطه بالا را طبق روش تکرار با جایگذاری، تکرار می‌کنیم، بنابراین:

$$\begin{aligned}T(n)&=3T\left(\frac{n}{4}\right)+n\\&=3^2T\left(\frac{n}{16}\right)+3\left(\frac{n}{4}\right)+n\\&=\ldots\\&\le 3^iT\left(\frac{n}{4^i}\right)+n\sum_{j=0}^{i-1}\left(\frac{3}{4}\right)^j\end{aligned}$$

رابطه بالا را تا زمانی که به $T(1)$ نرسیدیم ادامه می‌دهیم، بنابراین (با فرض اینکه n توانی از ۴ باشد):

$$\frac{n}{4^i}=1\Rightarrow i=\log_4^n$$

لذا خواهیم داشت:

$$T(n)\le 3^{\log_4^n}\times T(1)+n\sum_{j=0}^{\log_4^n-1}\left(\frac{3}{4}\right)^j$$

در عبارت بالا مقدار مجموع، برای nهای بزرگ، ثابت می‌باشد، لذا خواهیم داشت:

$$T(n)\le C_1n^{\log_4^3}+C_2n$$

که در آن C_1 برای $T(1)$ و C_2 برابر است با:

$$\lim_{n\to\omega}\sum_{j=0}^{\log_4^n-1}\left(\frac{3}{4}\right)^j=4\Rightarrow C_2<4$$

بنابراین در نهایت خواهیم داشت:

$$\begin{aligned}&T(n)\le C_1n^{\log_4^3}+4n\\&\Rightarrow T(n)\in O(n)\end{aligned}$$

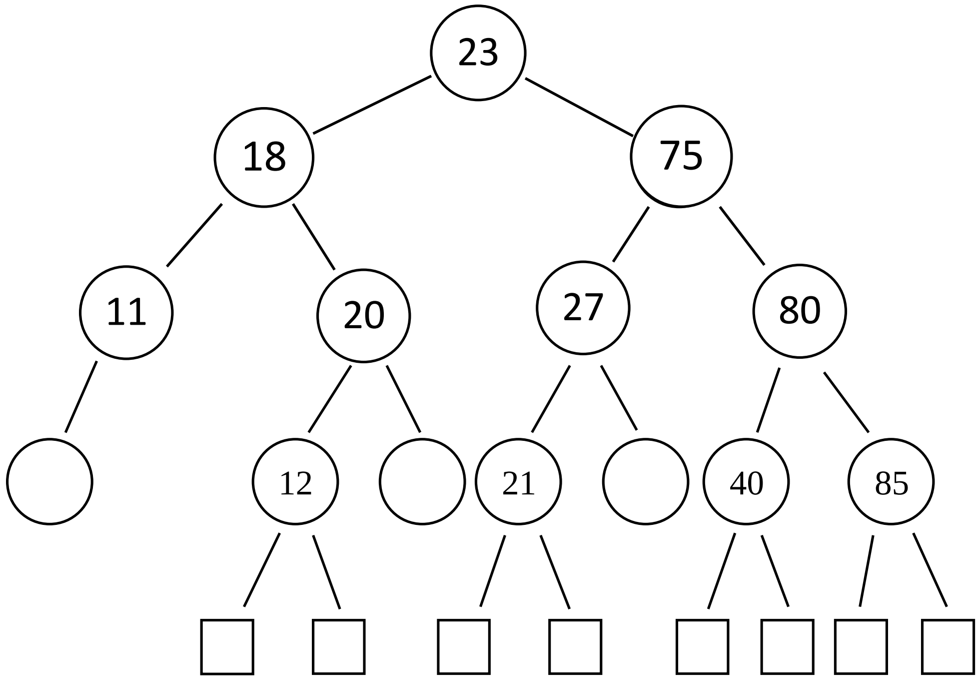
۱۱- اگر جستجوی دودویی را بر روی لیست زیر به دنبال عنصر 18 انجام دهیم، این عنصر در چندمین مقایسه یافته می‌شود؟

11,12,18,20,21,23,27,40,75,80,85

① 4 ② 3 ③ 2 ④ 1

پاسخ:

از آن جا که یک آرایه مرتب داریم، درخت جستجوی دودویی کامل را می‌شود به راحتی تشکیل داد. در نتیجه برای عناصر سطر اول ۱ مقایسه، برای عناصر سطر دوم ۲ مقایسه، سطر سوم ۳ مقایسه و ... لازم است.



۱۳- پیچیدگی زمانی الگوریتم مرتب سازی سریع در بدترین حالت کدام گزینه می‌باشد؟

(۱) $\theta(n)$ (۲) $\theta(n^2)$ (۳) $\theta(n\log n)$ (۴) $\theta(\log n)$

پاسخ:

[صفحه ۱۰۳ و ۱۰۴ کتاب] :

نخست باید بدترین شرایط را برای الگوریتم شناسایی کنیم. قبلاً اشاره کردیم که برای الگوریتمهایی که با روش تقسیم و حل طراحی می‌شوند، چنانچه داده‌ها و یا روش تقسیم داده‌ها به گونه‌ای باشد که داده‌ها در دو بخش با طول تقریباً یکسان قرار گیرند الگوریتم کارایی بهتری خواهد داشت. در الگوریتم QuickSort بدترین شرایط زمانی رخ می‌دهد که در مجموعه داده‌ها، هیچ دو یا چند مجموعه برابر وجود نداشته باشد و در هر بار فراخوانی partition، یک زیرمجموعه حاصل، تهی و زیرمجموعه دیگر شامل کلیه داده‌ها به استثنای عنصر محوری باشد. و این حالت زمانی رخ می‌دهد که این مجموعه داده‌ها از قبل مرتب شده باشند.

لذا هنگامی که partition در بالاترین سطح فراخوانی می‌شود هیچ عنصری در طرف چپ عنصر محوری قرار نمی‌گیرد و مقداری که برای pivotpiont توسط تابع partition ارسال می‌گردد برابر صفر است. به همین ترتیب در فراخوانی‌های بعدی نیز عنصر محور مقدار low را می‌گیرد. بنابراین لیست S به‌طور مکرر به یک زیرلیست خالی در طرف چپ و زیرلیست با یک عنصر کمتر در طرف راست تقسیم‌بندی می‌شود. لذا در بدترین حالت تابع زمانی را به‌صورت زیر می‌توان نوشت:

$$T(n) = \underbrace{T(0)}_{\text{زمان لازم برای مرتب‌سازی زیرلیست طرف چپ}} + \underbrace{T(n-1)}_{\text{زمان لازم برای مرتب‌سازی زیرلیست طرف راست}} + \underbrace{n-1}_{\text{زمان لازم برای تقسیم‌بندی لیست به دو زیرلیست}}$$

در ضمن یادآور می‌شوم که اعمال اصلی که در این الگوریتم صورت می‌گیرند عبارت است از مقایسه و نسبت‌دهی. و ما در این محاسبه عمل اصلی را مقایسه در نظر می‌گیریم.

بنابراین $T(n)$ در حالت کلی به‌صورت زیر خواهد بود:

$$T(n)=\begin{cases}0 & \text{if} \quad n<1\\ T(n-1)+n-1 & \text{if} \quad n\geq 1\end{cases} \qquad (4\text{-}7)$$

رابطه بازگشتی بالا را به روش‌های مختلف می‌توان حل کرد. ما روش تکرار و جایگذاری را برای حل به‌کار می‌بریم.

بنابراین خواهیم داشت:

$$\begin{aligned}T(n)&=T(n-1)+(n-1)\\ &=T(n-2)+(n-2)+(n-1)\\ &=T(n-3)+(n-3)+(n-2)+(n-1)\\ &\vdots\\ &=T(1)+1+2+\ldots+(n-2)+(n-1)\\ &=T(0)+1+2+\ldots+(n-2)+(n-1)\end{aligned}$$

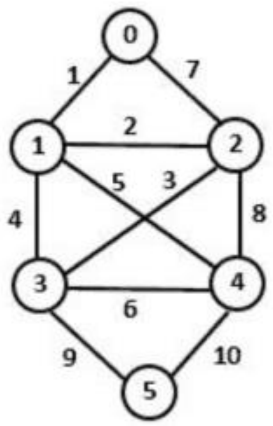
مجموع این جملات برابر خواهد بود با:

$$T(n)=\frac{n(n-1)}{2}$$

بنابراین پیچیدگی زمانی رابطه (۴-۷) برابر است با:

$$T(n)\in\theta\left(n^2\right)$$

۱۵- اگر به روش کروسکال درخت پوشای مینیمم را برای گراف شکل زیر به دست آوریم در مرحله سوم کدام یال به درخت اضافه می‌گردد؟



① یال 01 ② یال 02 ③ یال 14 **④ یال** 23

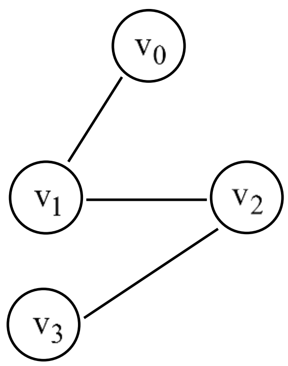
پاسخ:

نخست درخت heap برای مرتب کردن یال‌ها برحسب وزن یال‌ها تشکیل می شود، که در ریشه درخت heap ، $e_{01}=1$ قرار دارد و $F=\phi$ و Y مجموعه مجزا از هم تشکیل می‌شود. (توسط روال initial)

مرحله اول: $e_{01}=1$ به عنوان یال با هزینه کمتر انتخاب می‌شود که در آن $u=v_0$ و $v=v_1$ می‌باشد (یال e_{01} از درخت heap حذف می‌شود)، بنابراین داریم: $F=\{e_{01}\}$

مرحله دوم: $e_{12}=2$ به عنوان یال کم هزینه انتخاب می‌شود که در آن $u=v_1$ و $v=v_2$ می‌باشد (یال e_{12} از درخت heap حذف می‌شود)، بنابراین داریم: $F=\{e_{01},e_{12}\}$

مرحله سوم: $e_{23}=3$ به عنوان یال کم هزینه انتخاب می‌شود که در آن $u=v_2$ و $v=v_3$ می‌باشد (یال e_{23} از درخت heap حذف می‌شود)، بنابراین داریم: $F=\{e_{01},e_{12},e_{23}\}$



۱۷- مسئله ضرب زنجیره‌ای ماتریس‌ها را برای 4 ماتریس با ابعاد زیر در نظر بگیرید و مشخص کنید ترتیب بهینه ضرب که منجر به کمترین تعداد عمل ضرب می‌شود، کدام گزینه می‌باشد؟

$$A_{20\times 2}\times B_{2\times 30}\times C_{30\times 12}\times D_{12\times 8}$$

۱) $A(B(CD))$ ۲) $(AB)(CD)$ ۳) $A((BC)D)$ ۴) $(A(BC))D$

پاسخ:

بررسی گزینه‌ها:

گزینه ۱ $A(B(CD))=30\times 12\times 8+2\times 30\times 8+20\times 2\times 8=3680$

گزینه ۲ $(AB)(CD)=20\times 2\times 30+30\times 12\times 8+20\times 30\times 8=8880$

گزینه ۳ $A((BC)D)=2\times 30\times 12+2\times 12\times 8+20\times 2\times 8=1232$

گزینه ۴ $(A(BC))D=2\times 30\times 12+20\times 2\times 12+20\times 12\times 8=3120$

همانطور که ملاحظه می‌کنید، ترتیب گزینه ۳ برای ضــــرب این چهار ماتریس، ترتیب بهینه اســـت. ترتیب بهینه فقط به ابعاد ماتریس بستگی دارد.

۱۹- کدام گزینه صحیح نیست؟

۱) اغلب مسائلی که با تکنیک عقبگرد حل می‌شوند به شکلی هستند که از اصول و مفاهیم درخت‌ها استفاده می‌کنند.

۲) تکنیک عقبگرد حالت مصطلح شده‌ی جستجوی عمقی یک درخت می‌باشد.

۳) درخت تصمیم در تکنیک عقبگرد کاربردی ندارد و در سایر روش‌ها استفاده می‌شود.

۴) در تکنیک عقبگرد چنانچه مسئله بیش از یک جواب داشته باشد، همه جواب‌ها را باید پیدا کنیم.

پاسخ:

گزینه های ۱ و ۲ و ۴ صحیح هستند. [صفحه ۲۵۲ و ۲۵۳ کتاب]

گزینه ۳ غلط است:

[صفحه ۲۵۲ و ۲۵۳ کتاب] - مسائلی که به روش عقبگرد حل می‌شوند، اغلب مسائل تصمیم‌گیری هستند. در روش عقبگرد، مجموعه تصمیمات را به صورت یک درخت نمایش می‌دهند. چنین درختی را درخت تصمیم (Decision tree) می‌نامند.

۲۱- کدام یک از جملات زیر در مورد روش انشعاب و تحدید صحیح است؟

الف) فضای حالت مسئله‌ای که قرار است با این روش حل شود، باید با یک گراف قابل نمایش باشد.

ب) این روش، شکل بهبود یافته‌ای از روش تقسیم و حل می‌باشد.

ج) الگوی جستجو درخت در این روش به شکل ردیفی یا همان جستجو در پهنا می‌باشد.

① الف و ب　　② الف و ب و ج　　**③ الف و ج**　　④ فقط الف

پاسخ:

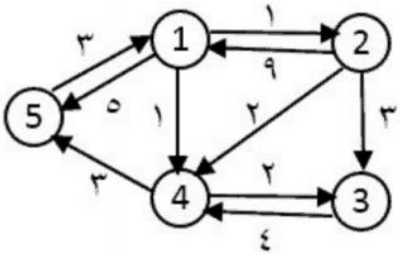
بررسی عبارت‌ها:

الف) درست : [صفحه ۳۰۱ کتاب]

ب) غلط ← انشعاب و تحدید روش اصلاح شده‌ی تکنیک عقبگرد است.

ج) درست : [صفحه ۳۰۱ کتاب] - الگوی جستجو در درخت برای روش بازگشت به عقب روش جستجوی عمقی است درحالیکه برای روش انشعاب و تحدید جستجوی ردیفی یا جستجو در پهنا می‌باشد.

۲۳- ماتریس همجواری گراف زیر کدام است؟



① $\begin{bmatrix} 0 & 1 & \infty & 1 & 5 \\ 9 & 0 & 3 & 2 & 5 \\ 0 & \infty & 0 & 4 & 0 \\ \infty & \infty & 3 & \infty & \infty \\ 3 & \infty & \infty & \infty & 0 \end{bmatrix}$

② $\begin{bmatrix} 0 & 1 & \infty & 1 & 5 \\ 9 & 0 & 3 & 2 & \infty \\ \infty & \infty & 0 & 4 & \infty \\ \infty & \infty & 2 & 0 & \infty \\ 3 & \infty & \infty & \infty & 0 \end{bmatrix}$

③ $\begin{bmatrix} 0 & 1 & \infty & 1 & 5 \\ 0 & 0 & 3 & 2 & \infty \\ \infty & \infty & 0 & 4 & \infty \\ \infty & 2 & 2 & 0 & \infty \\ 3 & \infty & \infty & \infty & 0 \end{bmatrix}$

④ $\begin{bmatrix} 0 & 1 & \infty & 1 & 2 \\ 0 & 0 & 3 & 2 & \infty \\ \infty & 3 & 0 & 4 & \infty \\ \infty & 2 & 2 & 0 & \infty \\ 5 & \infty & \infty & \infty & 0 \end{bmatrix}$

پاسخ:

می‌توان برای حل این سوال از روش رد گزینه استفاده کرد.

به این ترتیب می‌توانیم اختلافات گزینه ها را پیدا کنیم و بر اساس آن‌ها، جواب را بیابیم.

الف) اختلاف در $D[1][5]$: با توجه به گراف $D[1][5]=5$ ، بنابراین گزینه ۴ غلط است. $\{D[1][5]=5\neq 2\}$

ب) اختلاف در $D[1][5]$: با توجه به گراف $D[1][5]=9$ ، بنابراین گزینه ۳ غلط است. $\{D[1][5]=9\neq 0\}$

ج) اختلاف در $D[2][5]$: با توجه به گراف $D[2][5]=\infty$ (چون یال وجود ندارد) ، بنابراین گزینه ۱ غلط است.
$\{D[2][5]=\infty\neq 5\}$

روش دوم برای حل این سوال، به دست آوردن مستقیم ماتریس همجواری گراف داده شده است به این صورت که:

$$W[i][j]=\begin{cases}\text{وزن یال} & \text{اگر یالی بین } V_i \text{ و } V_j \text{ باشد:}\\ \infty & \text{اگر یالی بین } V_i \text{ و } V_j \text{ نباشد:}\end{cases}$$

$$\begin{array}{c c} & \begin{array}{ccccc}1 & 2 & 3 & 4 & 5\end{array}\\ \begin{array}{c}1\\2\\3\\4\\5\end{array} & \begin{bmatrix}0 & 1 & \infty & 1 & 5\\ 9 & 0 & 3 & 2 & \infty\\ \infty & \infty & 0 & 4 & \infty\\ \infty & \infty & 2 & 0 & \infty\\ 3 & \infty & \infty & \infty & 0\end{bmatrix}\end{array}$$

۲۵- برای کدام یک از مسائل زیر تا به حال الگوریتمی با مرتبه زمانی چند جمله ای پیدا نشده است؟

۱ مساله مرتب سازی یک لیست n عنصری
۲ مسئله جستجو یک آیتم در یک لیست n عنصری
۳ مسئله رنگ آمیزی گراف
۴ مسئله ضرب دو ماتریس $n\times n$

پاسخ:

مسئله فروشنده دوره‌گرد، مسئله n وزیر، **مسئله رنگ آمیزی گراف**، مسئله کوله پشتی و مسئله حاصل جمع زیرمجموعه‌ها جزو مسائلی هستند که تا به حال نتوانسته‌اند الگوریتمی با مرتبه زمانی چند جمله‌ای برای آن‌ها پیدا کنند.

برای این مسائل، الگوریتم‌های انشعاب و تحدید، الگوریتم‌های عقبگرد وجود دارد که برای بسیاری از نمونه‌ها بازدهی دارند، ولی الگوریتم‌های ارائه شده با این روش‌ها برای این مسائل از مرتبه نمائی می‌باشد و احتمال وجود الگوریتم‌های کاراتر رد نمی‌شود.

برای {گزینه ۱} مرتب سازی الگوریتم‌ها $O(n\log n)$، برای {گزینه ۲} جستجو در یک آرایه مرتب یک الگوریتم $O(\log n)$، برای {گزینه ۴} ضرب ماتریس‌ها یک الگوریتم $O\left(n^{2.38}\right)$ پیدا شده است.

سوالات تشریحی (تابستان ۹۶) / سوالات فرد

۱- رابطه بازگشتی زیر را با روش مناسب حل نموده و مرتبه اجرایی آن را مشخص کنید.

$$T(n)=\begin{cases}3T(n-1)+4T(n-2) & \text{if } n\ge 2\\ t(0)=0\ ,\ t(1)=1\end{cases}$$

پاسخ:

برای حل این رابطه، $T(n)=X^n$ قرار می‌دهیم. بنابراین:

$$X^n=3X^{n-1}+4X^{n-2}$$

با تقسیم طرفین بر X^{n-2}، معادله درجه دوم زیر حاصل می‌شود:

$$X^2=3X+4\ \Rightarrow\ X^2-3X-4=0$$

جواب‌های معادله عبارتند از: $X_1=-1$ و $X_2=4$

بنابراین:

$$T(n)=C_1X_1^n+C_2X_2^n\ \Rightarrow\ T(n)=C_1(-1)^n+C_2(4)^n$$

با توجه به شرایط مرزی $T(0)$ و $T(1)$ در معادله، داریم:

$$C_1+C_2=0$$
$$\underline{-C_1+4C_2=1}$$
$$\Rightarrow 5C_2=1\Rightarrow \boxed{C_2=\frac{1}{5}}\Rightarrow \boxed{C_1=-\frac{1}{5}}$$

بنابراین:

$$T(n)=-\frac{1}{5}(-1)^n+\frac{1}{5}(4)^n\ \Rightarrow\ T(n)=\frac{1}{5}\left(4^n-(-1)^n\right)$$

و در نتیجه:

$$T(n)\in O\left(4^n\right)$$

۳- ۷ کار طبق جدول زیر با بهره و مهلت معین وجود دارد. با فرض اینکه زمان انجام همه کارها یکسان و برابر با یک واحد زمانی می‌با شد، با روش حری صانه یک ترتیب بهینه برای انجام کارها به گونه‌ای ارائه دهید که بی شترین سود حاصل شود.

کار	مهلت	بهره
۱	۳	۶۰
۲	۱	۵۰
۳	۱	۳۰
۴	۲	۲۰
۵	۳	۱۵
۶	۱	۱۰
۷	۲	۵

پاسخ:

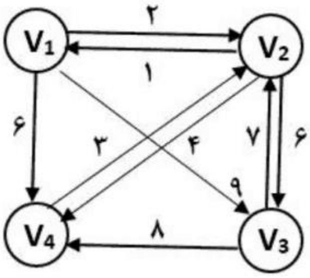
مقدار j ابتدا برابر صفر است. جدول زیر مراحل اجرا را نشان می‌دهد:

مرحله	j	سود	امکان پذیر بودن مجموعه
0	0	0	هست
1	$\{1\}$	60	هست
2	$\{2,1\}$	110	هست
3	$\{2,3,1\}$	140	نیست
4	$\{2,4,1\}$	130	هست
5	$\{2,4,1,5\}$	145	نیست
6	$\{1,2,6,4\}$	140	نیست
7	$\{2,4,7,1\}$	135	نیست

مثلا زمانبندی $\{2,3,1\}$ امکان پذیر نیســـت، زیرا کار 2 در زمان 1 آغاز شـــده و یک واحد زمانی طول می‌کشـــد و این موجب می‌شود که کار 3 در زمان 2 شروع شود، در حالی که آخرین مهلت کار 3 برابر 1 است.

در نتیجه، جواب بهینه $j=\{2,4,1\}$ با سود 130 خواهد بود.

۵- گراف شکل زیر را در نظر بگیرید و با استفاده از روش برنامه نویسی پویا یک تور بهینه برای آن بیابید. (مسئله فروشنده دوره گرد)



پاسخ:

با توجه به ماتریس مجاورتی این گراف یک تور بهینه بیان می‌کنیم:

$D[V_2][0]=1$

$D[V_3][0]=\infty$

$D[V_4][0]=6$

ابتدا مجموعه تهی را در نظر می‌گیریم:

یعنی A را برابر با V_1 در نظر می‌گیریم و مســافت هر رأس تا V_1 را پیدا می‌کنیم. با توجه به ماتریس مجاورتی چون از V_3 به V_1 یالی وجود ندارد، از علامت ∞ استفاده می‌کنیم.

حال همه مجموعه‌های حاوی یک عنصر را در نظر می‌گیریم:

$$D[V_3][\{V_2\}]=\min imum=W[3][j]+D\left[V_j\right]\left[\{V_2\}-\{V_j\}\right]$$
$$V_j\in\{V_2\}=W[3][2]+D[V_2][0]=7+1=8$$

به طور مشابه:

$D[V_4][\{V_2\}]=3+1=4$

$D[V_2][\{V_3\}]=6+\infty=\infty$

$D[V_4][\{V_3\}]=\infty+\infty=\infty$

$D[V_2][\{V_4\}]=4+6=10$

$D[V_3][\{V_4\}]=8+6=14$

سپس همه مجموعه‌های حاوی دو عنصر را در نظر می‌گیریم:

$$D[V_4][\{V_2,V_3\}]=\min imum=\left(W[4][j]+D\left[V_j\right]\left[\{V_2,V_3\}-\{V_j\}\right]\right)$$
$$V_j\in\{V_2,V_3\}$$
$$=\min imum=\left(W[4][2]+D[V_2][\{V_3\}],W[4][3]+D[V_3][\{V_2\}]\right)$$
$$\min imum=(3+\infty,\infty+8)=\infty$$

و به طور مشابه:

$D[V_3][\{V_2,V_4\}]=\min imum=(7+10,8+4)=12$

$D[V_2][\{V_3,V_4\}]=\min imum=(6+14,4+\infty)=20$

در نهایت، طول یک تور بهینه را به صورت زیر محاسبه می‌کنیم:

$$D[V_1][\{V_2,V_3,V_4\}]=\min imum=\left(W[1][j]+D\left[V_j\right]\left[\{V_2,V_3,V_4\}-\{V_j\}\right]\right)$$
$$V_j\in\{V_2,V_3,V_4\}$$
$$=\min imum=\left(W[1][2]+D[V_2][\{V_3,V_4\}],W[1][3]+D[V_3][\{V_2,V_4\}],W[1][4]+D[V_4][\{V_2,V_3\}]\right)$$
$$\min imum=(2+20,9+12,\infty+\infty)=21$$

۲- یک آرایه از اعداد صحیح به صورت $A[1...n]$ مفروض است طوری که $\sum_{i=1}^{n} A[i]$ می‌باشد، در این صورت مرتبه اجرایی الگوریتم زیر کدام است؟

```
Ax = o
For(i = o;i < n;i++)
For(j=1;j<=[i];j++)
X++
```

① $o(m)$ ۲ $o(n)$ ③ $o(nm)$ ④ $o(n+m)$

پاسخ:

سطر	هزینه	تعداد
1	C_1	1
2	C_2	$n-1$
3	C_3	$(m-1)+(m-2)+...+(m-n+1)$
4	C_4	$(m-2)+...+(m-n)$

پس هزینه کل برابر است با:

$$T(m,n)=C_1+C_2(n-1)+C_3((m-1)+(m-2)+...+(m-n+1))+C_4((m-2)+...+(m-n))$$

با انتخاب $C=\max\{C_1,C_2,C_3,C_4\}$ داریم:

$$T(n,m)\in O(n+m)$$

بنابراین گزینه ④ صحیح می‌باشد.

۴- با استفاده از کدام یک از روابط زیر میتوان رابطه $f(n)\in\theta(g(n))$ را نتیجه گیری نمود؟

① $\lim_{x\to\infty}\frac{f(n)}{g(n)}=0$ ② $f(n)\in O(g(n))$ و $g(n)\in O(f(n))$

③ $\lim_{x\to\infty}\frac{f(n)}{g(n)}=10^5$ ④ $f(n)\in O(h(n))$ و $h(n)\in g(n)$

پاسخ:

① طبق قضیه ۴-۱ از این رابطه نتیجه می‌شود: $T(n)\in O(g(n))$ و $T(n)\notin O(g(n))$ پس این گزینه نادرست است.

② طبق قضیه ۳-۱ این رابطه نادرست است. زیرا داریم: $g(n)\in\Omega(f(n))$ اگر و فقط اگر $f(n)\in\theta(g(n))$

③ طبق قضیه ۴-۱ از این رابطه نتیجه می‌شود که $f(n)\in\theta(g(n))$ پس این گزینه صحیح می‌باشد.

④ طبق قضیه ۳-۱ این رابطه نادرست است زیرا داریم: $T(n)\in\theta(g(n))$ و $g(n)\in\theta(h(n))\to T(n)\in\theta(h(n))$

۶- آرایه زیر را در نظر بگیرید خروجی تابع زیر را به ازای $f(a,9)$ چیست؟

$\text{Int } a[10]=\{1,2,3,4,5,6,7,8,9,10\}$

$\text{Int } f(\text{int } a, \text{int } n)$

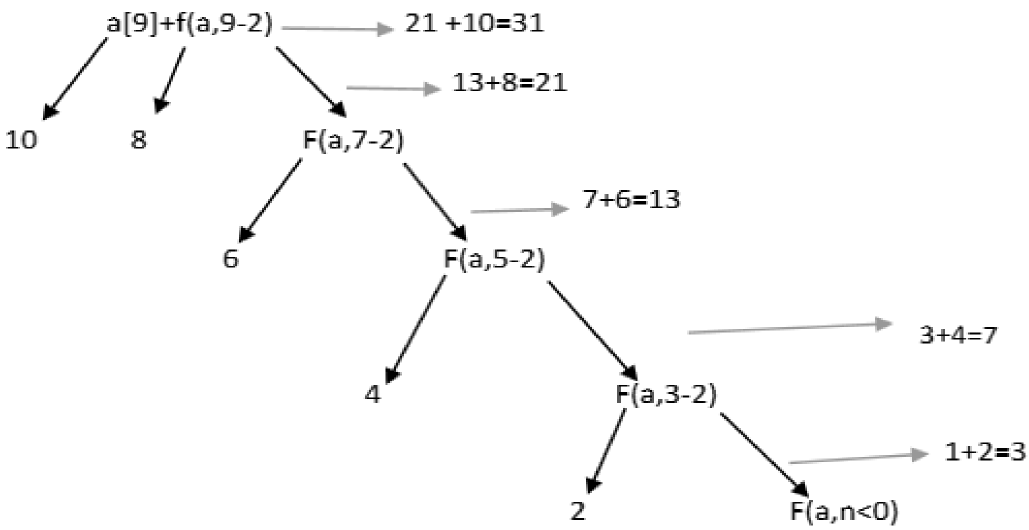
$\text{If } (n<0) \text{return1;}$

$\text{Return } a[a]+f(a,n-2);$

۱) 25 ۲) 30 ۳) 31 ۴) 55

پاسخ:

گزینه ۳ صحیح است زیرا داریم:



۸- زمان اجرای تابع بازگشتی زیر کدام است؟

```
int f (int n)
if (n <= 2) return1;
return f (n − 2) * f (n − 2);
```

① $O(n^2)$ ② $O(2^n)$ ③ $O(n\log n)$ ④ $O\left(2^{\frac{n}{2}}\right)$

پاسخ:

(تابع فیبوناتچی است)

$$T(n) > 2\times T(n-2)$$
$$> 2\times 2\times T(n-4)$$
$$> 2\times 2\times 2\times T(n\times 6)$$
$$\dots$$
$$> \underbrace{2\times 2\times 2\times \dots\times 2}_{\frac{n}{2}\ \text{مرتبه}}\times T(0)$$

با توجه به اینکه $T(0)=1$ است پس $T(n)>2^{n/2}$ می‌شود. اثبات:

پایه استقرا: رابطه فوق به ازای $n=2$ و $n=3$ درست است چرا که $\text{fib}(2)$ سه بار به صد زدن تابع نیاز دارد:

$$T(2)=3>2=2^{2/2}$$
$$T(3)=5>2^{3/2}$$

و $\text{fib}(3)$ به 5 بار صدا زدن تابع نیاز دارد:

فرض استقرا: فرض می‌کنیم رابطه $T(n)>2^{n/2}$ به ازای همه مقادیر m کوچک‌تر از n درست است.

حکم استقرا: ثابت می‌کنیم به ازای n نیز درست می‌باشد. از روی تعریف مشخص است که $T(n)$ برابر است با:

$$T(n)=T(n-1)+T(n-2)+1$$
$$> 2^{(n-1)/2}+2^{(n-2)/2}+1$$
$$> 2^{(n-2)/2}+2^{(n-2)/2}=2*2^{(n/2)-1}=2^{n/2}$$
$$\rightarrow T(n)>2^{n/2}$$

حکم استقرا ثابت شد.

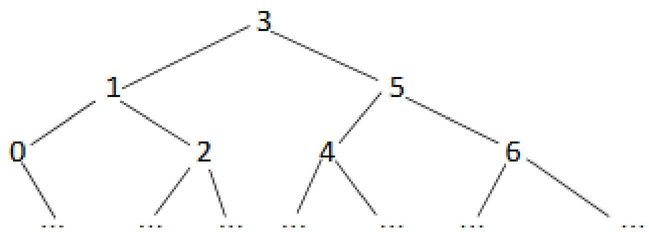
گزینه ④ صحیح است.

۱۰- **در الگوریتم جست و جوی دودویی متوسط تعداد مقایسه‌ها در جست و جو ناموفق برای یک آرایه 7 عنصری چقدر است؟**

۱) $\frac{17}{7}$ ۲) 3 ۳) $\frac{17}{8}$ ۴) 4

پاسخ:

با توجه به درخت جست و جوی دودویی داریم:



$$\text{متوسط تعداد مقایسه در جست و جوی ناموفق} = (1+7*3)/8 = 3$$

گزینه ۲ صحیح است.

۱۲- **برای دو ماتریس 8*8 با دو روش برنامه نویسی پویا و استراسن چه تعداد عمل ضرب نیاز است؟ فرض کنید کوچک در روش استراسن ضرب ماتریس 2*2 باشد.**

۱) برنامه نویسی پویا = ۵۱۲ استراسن = ۳۴۳
۲) برنامه نویسی پویا = ۶۴ استراسن = ۴۹
۳) **برنامه نویسی پویا = ۵۱۲ استراسن = ۳۹۲**
۴) برنامه نویسی پویا = ۶۴ استراسن = ۳۹۲

پاسخ:

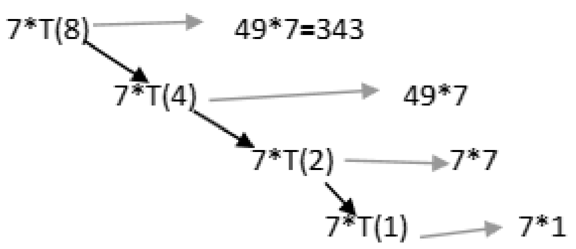
به طور کلی برای ضرب یک ماتریس $i\times j$ در یک ماتریس $i\times k$ با استفاده از ضرب استاندارد ماتریس‌ها در برنامه نویسی پویا تعداد اعمال ضرب لازم $i\times j\times k$ خواهد بود یعنی داریم: $8\times8\times8=512$

در الگوریتم استراسن هنگامی که دو ماتریس $n\times n$ با $n>1$ داشته باشیم دقیقا هفت بار فراخوانی می‌شود و در هر بار که ماتریس $(n/2)*(n/2)$ ارسال می‌شود هیچ ضربی در بالاترین سطح انجام نمی‌پذیرد.

بنابراین تعداد عمل ضرب برابر است با:

$$T(n)=\begin{cases}7T\left(\frac{n}{2}\right) & \text{if } n>1\\ 1 & \text{if } n\le 1\end{cases}$$

پس داریم:



گزینه ۱ صحیح است.

۱۴- زمان اجرای هر کدام ازکارهای زیر یک واحد زمانی است هر کار دارای سود و مهلت معینی می‌باشد که اگر بعد مهلت داده شده انجام شود سودی به آن تعلق نمی‌گیرد کارهای زیر را با چه ترتیبی از چپ به راست انجام شود تا سود حاصل شده حداکثرشود ؟

کار	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲
مهلت	۱	۷	۶	۴	۲	۶	۴	۳	۲	۴	۷	۵
سود	۱۹	۷	۲۵	۱۵	۶	۱۶	۲۷	۱۰	۳	۲۳	۴	۲

۱) 1,7,3,10,6,11,2 ۲) 1,3,7,10,6,2,11 ۳) 1,7,3,10,6,2,11 ۴) 1,3,7,10,6,11,2

پاسخ:

با استفاده از روش زمان‌بندی با مهلت نخست کار با اولویت بالا یعنی کاری که مهلت آن کم است (در این مساله کار شماره 1 است) را انجام دهیم سپس کار با اولویت بالا را انجام شود. پس کار ها را بر اساس اولویت سود به صورت غیر نزولی مرتب کرده و با استفاده از روش حریصانه انتخاب می‌شود که چه کاری را باید زودتر انجام داد.

کار	۱۲	۹	۱۱	۵	۲	۸	۴	۶	۱	۱۰	۳	۷
سود	۲	۳	۴	۶	۷	۱۰	۱۵	۱۶	۱۹	۲۳	۲۵	۲۷

رد گزینه:

سود کار ۷ بیش‌تر از ۳ است ⇐ گزینه ۲ و ۴ نادرست است.

سود کار ۲ بیش‌تر از کار ۱۱ است ⇐ گزینه ۳ صحیح است.

۱۶- برای ضرب دو جمله‌ای با استفاده از الگوریتم زیر چه تعداد عمل جمع مورد نیاز است؟

```
int b(int n,int k)
if (k == 0| |n == k)return1;
Return b(n-1,k-1)+b(n-1,k);
```

① $\binom{n}{k}-1$ ② $2\binom{n}{k}-2$ ③ $2\binom{n}{k}-1$ ④ $\frac{k(2n-k-1)}{2}$

پاسخ:

این الگوریتم محاسبه ضرب دو جمله‌ای با استفاده از تقسیم و حل می‌باشد. هرعامل باید آن قدر بسط داده شود که در نهایت به 1 برســد و یک‌ها را باید با هم جمع کرد. قطعا باید $\binom{n}{k}$ عدد را با هم جمع کنیم که $\binom{n}{k}$ شــود، در این صــورت تعداد اعمال جمع $\binom{n}{k}-1$ خواهد بود.

۱۸- فرض کنید قرار ا ست بین دو تیم A و B یک م سابقه انجام شود. اولین تیمی که تعداد بردش n شود برنده م سابقه ا ست. با فرض اینکه بازی نتیجه م ساوی ندا شته با شد حداکثر تعداد بازی‌هایی که باید برگزار شود چقدر است؟

① n ② $n+1$ ③ $2n-1$ ④ $2n$

پاسخ:

چون ممکن اســت یک در میان، تیمی بر تیم دیگر غلبه کند وقتی که اولین تیم به n برد می‌رســد تیم دوم $n-1$ برد خواهد داشت پس حداکثر تعداد مسابقه‌ای که برگزار می‌شود $2n-1$ بازی خواهد بود.

بنابراین گزینه ج صحیح است.

۲۰- تعداد درخت‌ها جست و جو دودویی با n کلید متمایز کدام است؟

① $\frac{1}{n+1}\binom{2(n-1)}{n-1}$ ② $\frac{1}{n+1}\binom{2n}{n}$ ③ $\frac{1}{n}\binom{2(n-1)}{n-1}$ ④ $\frac{1}{n}\binom{2n}{n}$

پاسخ:

چون n راس و هر راس دو فرزند می‌تواند داشـــته باشـد پس در کل به تعداد $2n$ تا جای داریم. در این $2n$ تا جای باید n را کلید قرار دهیم پس به تعداد $\binom{2n}{n}$ حالت می‌توان کلیدها را قرار داد ولی به سادگی قابل رویت است که هر $n+1$ مدل چیدمان تنها یکی از آن‌ها خاصیت درخت‌های جستجو دودویی را دارد. پس تعداد کل درخت‌های جستجوی دودویی برابر است با:

$$\frac{1}{n+1}\binom{2n}{n}$$

۲۲- کدام گزینه درست است؟

① روش عقبگرد می‌تواند مرتبه زمانی مسائل سخت را کاهش دهد.

② روش عقبگرد اصلاح شده جست و جو ردیفی یک درخت است.

③ **مسائلی که به روش عقبگرد حل می‌شوند اغلب مسائل تصمیم‌گیری هستند.**

④ در روش عقبگرد تنها بهینه‌ترین جواب از میان جواب‌های مساله بدست می‌آید.

پاسخ:

گزینه ① : روش عقبگرد نمی‌تواند مرتبه زمانی مسایل سخت را کاهش دهد و فقط با کاهش حالت‌ها تنها زمان اجرا برای طول داده (n) کوچک کاهش می‌دهد. $\Leftarrow$ غلط

گزینه ② : روش عقبگرد اصلاح شده روش عمقی است. $\Leftarrow$ غلط

گزینه ④ : در روش عقبگرد چنانچه مسئله بیش از یک جواب داشته باشد، همه جواب‌ها را باید پیدا کنیم. $\Leftarrow$ غلط

۲۴- تعداد کل گره‌های درخت فضای حالت برای الگوریتم یافتن مدارهای همیلتونی با استفاده از تکنیک عقبگرد در کدام گزینه آمده است؟

① $\frac{n(n-1)}{n+1}$ ② $(n-1)^n-1/n-2$

③ $(n-1)^n+1/n+1$ ④ $(n-1)^{2n}+1/n-2$

پاسخ:

تعداد گره‌های درخت فضای حالت برای این الگوریتم عبارت است از:

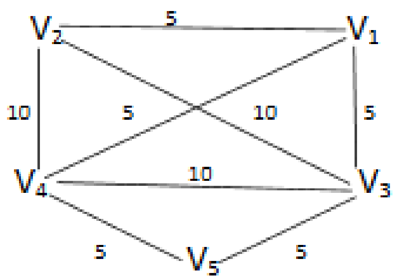
$$1+(n-1)+(n-2)^2+...+(n-1)^{n-1}=(n-1)^n-1/n-2$$

۲- یک صفحه مدار چاپی دارای 5 حفره است می‌خواهیم توسط نوارهای مسی این 5 حفره را با هم متصل باشند یعنی از هر حفره مسیری به حفره دیگر موجود باشد اگر حفره iام را به حفره jام متصل کنیم به $T[i][j]$ واحد طولی نوار مسی نیاز داریم چگونه این کار را انجام دهیم تا کم‌ترین طول نوار مسی مصرف شود. الگوریتم مورد استفاده برای حل این مساله و مراحل محاسبه جواب را به طور کامل بنویسید؟

$$T = \begin{bmatrix} 0 & 5 & 5 & 5 & \infty \\ 5 & 0 & 10 & 10 & \infty \\ 5 & 10 & 0 & 10 & 5 \\ 5 & 10 & 10 & 0 & 5 \\ \infty & \infty & 5 & 5 & 0 \end{bmatrix}$$

پاسخ:

با استفاده از گراف و الگوریتم کروسکال داریم:



با استفاده از روش کروسکال ابتدا کوچک‌ترین یال را که در اینجا از میان یال‌های $\{E_{12}, E_{13}, E_{14}, E_{35}, E_{54}\}$ یکی را انتخاب می‌کنیم. به طور مثال E_{13} را انتخاب کرده و سپس از V_3 به بقیه راس‌ها یال کوچک‌تر را که E_{35} است و پس از آن از یال E_{54} به V_7 رفته و بعد از آن چون هر یال از V_4 به V_2 ایجاد دور می‌کند ما کوچک‌ترین یال به V_2 که یال E_{12} است را انتخاب می‌کنیم پس مسیر نوار به این صورت است:

$$F = \{E_{13}, E_{35}, E_{54}, E_{12}\}$$

محاسبه طول نوار برابر است با جمع طول یال‌های مسیر:

5+5+5+5=20

۴- با استفاده از روش عقبگرد برای مساله کوله پشتی صفر و یک سود ماکزیمم قابل حصول از نمونه زیر پیدا کنید. عملیات را مرحله به مرحله نمایش دهید؟

h = 5 w = 40

i	p_i	w_i
1	8	16
2	10	8
3	5	15
4	15	25
5	20	15

پاسخ:

روش حل این سوال:

برای حل مسأله کوله‌پشتی توسط الگوریتم عقبگرد، ابتدا قطعات را براساس P_i/W_i به ترتیب غیرنزولی مرتب می‌کنیم که در آن W_i و P_i به ترتیب وزن و ارزش قطعه i هستند. فرض کنید می‌خواهیم امیدبخش بودن یک گره خاص را تعیین کنیم. برای این کار تعریف‌های زیر را در نظر بگیرید:

profit: حاصل جمع ارزش قطعاتی می‌باشد که تا آن گره در نظر گرفته شده‌اند.

weight: حاصل جمع اوزان قطعاتی می‌باشد که تا آن گره در نظر گرفته شده‌اند.

bound: حد بالایی از بهره قابل دستیابی، با استفاده از گسترش دادن گره است و به آن مقدار اولیه profit نسبت داده می‌شود.

total weight: نشان‌دهنده وزن کلی قطعات بوده و بـه آن مقـدار اولیـه weight نسبت داده می‌شود.

روش کار بدین صورت است که قطعات را براساس P_i/W_i آنها انتخـاب کـرده، سپس ارزش آنها را به bound و وزن آنها را بـه total weight اضـافه مـی‌کنیم. تـا بـه

قطعه‌ای برسیم که اگر وزن آن را به مجموع وزن‌های قبلی اضافه کنیم، total weight از W بیشتر شود.

فرض کنید گرهی در سطح i قـرار دارد و گـره واقـع در سـطح k (k>i)، گرهـی است که حاصل جمع اوزان را از W بیشتر کند. در این صورت خواهیم داشت:

$$\text{total weight} = \text{weight} + \sum_{j=i+1}^{k-1} W_j$$

$$\text{bound} = \underbrace{\left(\text{profit} + \sum_{j=i+1}^{k-1} P_j\right)}_{\text{بهره حاصل از (k-1) قطعه اول}} + \underbrace{(W - \text{total weight})}_{\text{اندازه ظرفیتی که قطعه k می‌تواند استفاده کند}} \times \underbrace{\frac{P_k}{W_k}}_{\text{ارزش واحد وزن قطعه kام}}$$

و اگر maxprofit مقدار بهره در بهترین حلی باشد که تاکنون پیدا شده اسـت، در این صورت یک گره در سطح i، غیرامیدبخش خواهد بود هرگاه داشته باشیم:

$$\text{bound} \le \text{max profit}$$

{مشابه مثال ۵-۷ کتاب – صفحه ۲۷۴}

مثال ۵-۷: فرض کنید n=٤ و W=١٦ داشته باشیم:

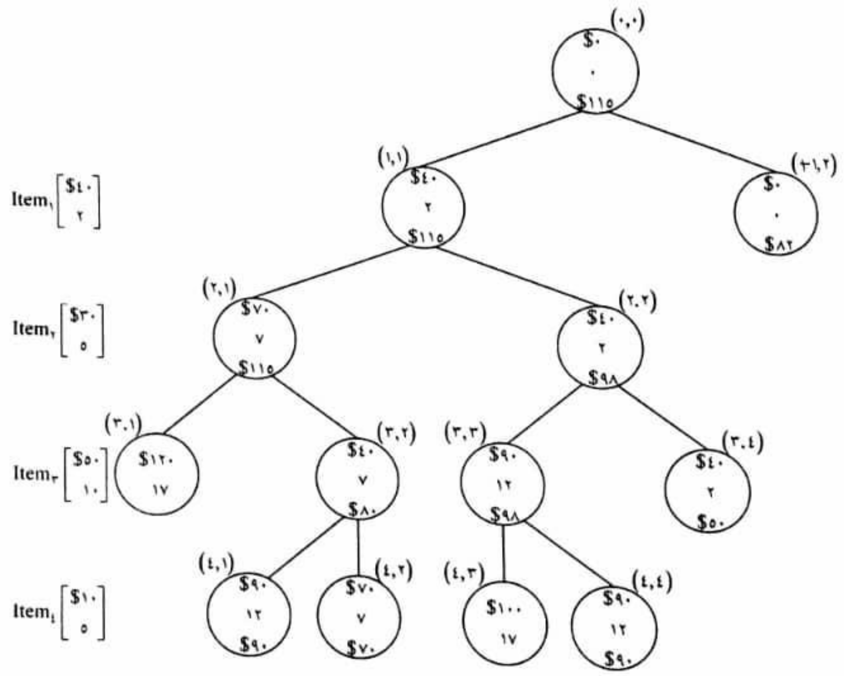
i	P_i	W_i	P_i/W_i
1	\$50	10	\$5
2	\$30	5	\$6
3	\$10	5	\$2
4	\$40	2	\$20

در ابتدای کار **قطعه‌های فوق** را براساس P_i/W_i به‌صورت غیرنزولی مرتب کنیم:

i	P_i	W_i	P_i/W_i
1	\$40	2	\$20
2	\$30	5	\$6
3	\$50	10	\$5
4	\$10	5	\$2

شکل ۱۲-۷ درخت فضای حالت هرس‌شده‌ای را نشان می‌دهد که از بکارگیری روش عقبگرد بدست آمده است. ارزش کل (profit)، وزن کل (weight) و حد (bound) در هر گره از بالا به پایین مشخص شده است. راه‌حل (جواب ماکزیمم یا جواب بهینه) در گرهی که به‌صورت سایه رنگ‌آمیزی شده است می‌باشد، همچنین فرض کرده‌ایم سطح ریشه صفر می‌باشد و فرزندان ریشه در سطح یک قرار دارند و غیره.

حال مراحلی که منجر به درخت فضای حالت هرس شده برای این مثال گشته است را توضیح می‌دهیم:



شکل ۱۲-۷: درخت فضای حالت هرس شده که جواب در گره (۳-۳) پیدا شده است.

۱. در ابتدای کار maxprofit=\$۰ قرار می‌دهیم.

۲. از گره ریشه شروع می‌کنیم و آن را بازدید می‌دهیم.

الف) ارزش و وزن آن را محاسبه می‌کنیم.

Profit =\$0
Weight = 0

ب) حد آن را محاسبه می‌کنیم.

برای محاسبه حد، ابتدا به محاسبه k می‌پردازیم. k نشان‌دهنده شماره سطحی است که اگر آن گره را به جمع وزن گره‌های قبلی اضافه کنیم حاصل جمع بدست آمده از W بیشتر خواهد شد.

چون ۱۷=۱۰+۵+۲ و ۱۷ از ۱٦ یعنی مقدار W بزرگتر است پس قطعه سوم حاصل جمع اوزان را از W بیشتر می‌کند. بنابراین k=۳ است و داریم:

$$\text{total weight} = \text{weight} + \sum_{j=i+1}^{3-1} W_j = 0 + 2 + 5 = 7$$

(i در سطح صفر، برابر صفر است).

$$\text{bound} = \text{profit} + \sum_{j=i+1}^{3-1} P_j \Big) + \left(W - \text{total weight}\right) \times \frac{P_3}{W_3}$$

$$= \$0 + \$40 + \$30 + \left(16 - 7\right) \times \frac{\$50}{10} = \$115$$

ج) چون وزن گره صفر است و آن کمتر از وزن تحملی کوله‌پشتی یعنی ۱٦ است. پس آن گره امیدبخش است حد گره ۱۱۵$ بزرگتر از $0، یعنی مقدار maxprofit است.

۳. گره (۱،۱) را بازدید می‌کنیم.

الف) ارزش و وزن آن را محاسبه می‌کنیم.

Profit= $0+$40=$40 و Weight = 0+2=2

ب) چون وزن آن ۲ است و کوچکتر از W=۱٦ است و ارزش آن $40 است بزرگتر از $0 (مقدار (maxprofit)است. maxprofit برابر $40 قرار داده می‌شود.

ج) حد آن را محاسبه می‌کنیم. چون ۱۷=۱۰+۵+۲ است و ۱۷ از W=۱٦ بزرگتر است. قطعه سوم حاصل جمع اوزان را از W فراتر می‌برد. بنابراین k=۳ است. بنابراین داریم:

$$\text{total weight} = \text{weight} + \sum_{j=i+1}^{3-1} W_j = 2 + 5 = 7$$

$$\text{bound} = \text{profit} + \sum_{j=i+1}^{3-1} P_j + \left(W - \text{total weight}\right) \times \frac{P_3}{W_3}$$

$$= \$40 + \$30 + \left(16 - 7\right) \times \frac{\$50}{10} = \$115$$

د) همانطور که ملاحظه می‌کنید، وزن آن ۲ است و کمتر از ۱٦ و حد آن که $115 است بیش از مقدار maxprofit یعنی $40 است.

٤. گره (۱،۲) را بازدید می‌کنیم.

الف) ارزش و وزن آن را محاسبه می‌کنیم.

Profit= $40+$30=$70 و Weight =2+5=7

ب) چون وزن آن ۷ است، بنابراین کوچکتر از ۱۶ می‌باشد و ارزش آن $70 می‌باشد که بزرگتر از مقدار maxprofit یعنی $40 است. مقدار maxprofit را به $70 تغییر می‌دهیم.

ج) حد آن را محاسبه می‌کنیم.

$$\text{total weight} = \text{weight} + \sum_{j=i+1}^{3-1} W_j = 7 \qquad \text{(i=۲ می‌باشد)}$$

$$\text{bound} = \$70 + (16-7) \times \frac{\$50}{10} = \$115$$

د) چون وزن گره که ۷ است کمتر از ۱۶ می‌باشد و حد آن که $115 است از مقدار maxprofit که $70 می‌باشد، کمتر است، پس گره امیدبخش است.

۵. گره (۳،۱) را بازدید می‌کنیم.

الف) ارزش و وزن آن را محاسبه می‌کنیم.

Profit= $70+$50=$120 و Weight =7+10=17

ب) چون وزن آن ۱۷ است و از ۱۶ بزرگتر است پس مقدار maxprofit تغییر نمی‌کند.

ج) گره امیدبخش نیست زیرا وزن آن که ۱۷ است بزرگتر از ۱۶ است.

د) حد محاسبه می‌شود، زیرا وزن آن، گره را امیدبخش کرده است.

۶. عقبگرد به گره (۲،۱)

۷. گره (۳،۲) را بازدید می‌کنیم.

الف) ارزش و وزن آن را محاسبه می‌کنیم. قطعه ۳ را لحاظ نمی‌کنیم.

Profit=$70 و Weight =7

ب) چون ارزش قطعه‌ای که $70 است، کوچکتر یا مساوی $70 می‌باشد، بنابراین، مقدار maxprofit تغییر نمی‌کند.

ج) حد آن را محاسبه می‌کنیم. وزن چهارم باعث نمی‌شود که حاصل جمع قطعات از W بالاتر رود و فقط چهار قطعه وجود دارد. بنابراین k=۵ است وخواهیم داشت:

$$\text{bound} = \text{profit} + \sum_{j=i+1}^{5-1} P_j = \$70 + \$10 = \$80 \qquad (i=3)$$

د) چون وزن گره که ۷ است کوچکتر از ۱۶ بوده و حد آن $80 بوده و بزرگتر از مقدار maxprofit می‌باشد. پس گره امیدبخش است.

(از این به بعد، محاسبات مربوط به ارزش‌ها، اوزان و حدها را به‌عنوان تمرین به عهده دانشجو می گذاریم، و هنگامی که maxprofit تغییری نمی‌کند، آن را ذکر نمی‌کنیم.)

۸. گره (۱،٤) را بازدید می‌کنیم.
الف) ارزش وزن آن را برابر $80 و ۱۲ قرار می‌دهیم.
ب) چون وزن آن که ۱۲ است کوچکتر از ١٦ می‌باشد و ارزش آن که $80 است بزرگتر از $70 می‌باشد maxprofit به $80 تغییر می‌کند.
ج) حد برابر $80 محاسبه می‌شود.
د) چون حد آن که $80 است کوچکتر یا مساوی مقدار maxprofit که $80 است می‌باشد، پس گره غیرامیدبخش است.
۹. به گره (۳،۲) عقبگرد می‌کنیم.
۱۰.گره (٤،۲) را بازدید می‌کنیم.
الف) ارزش و وزن آن را که برابر $70 و ۷ می‌باشد، محاسبه می‌کنیم.
ب) حد آن $70، محاسبه می‌شود.
ج) چون حد آن که $70 است کوچکتر یا مساوی $80 می‌باشد پس گره امیدبخش است.

۱۱.عقبگرد به گره (۱،۱).
۱۲.گره (۲،۲) را بازدید می‌کنیم.
الف) ارزش و وزن آن که برابر $400 و ۲ می‌باشد را محاسبه می‌کنیم.
ب) حد آن $98 محاسبه می‌شود.
ج) چون وزن آن ۲ کمتر از ١٦ است و حد آن که $98 است بزرگتر از $80 (مقدار maxprofit) می‌باشد. پس گره امیدبخش است.
۱۳.گره (۳،۳) را بازدید می‌کنیم.
الف) ارزش و وزن آن را که برابر $90 و ۱۲می‌باشد، را محاسبه می‌کنیم.
ب) چون وزن آن که ۱۲ است کوچکتر از ١٦ می‌باشد و ارزش آن که $90 است بیش از مقدار کنونی maxprofit که $80 می‌باشد. مقدار maxprofit را به $90 تغییر می‌دهیم.
ج) حد آن را برابر $90 محاسبه می‌کنیم.
د) گره امیدبخش است.
١٤.عقبگرد به گره (٤،۳)
الف) ارزش و وزن آن را که برابر $100 و ۱۷ می‌باشد را محاسبه می‌کنیم.
ب) وزن گره ۱۷ بوده، بیشتر از ١٦ می‌باشد، پس گره امیدبخش نیست.
۱٥.عقبگرد به گره (۳،۳)
١٦.گره (٤،٤) را بازدید می‌کنیم.
الف) ارزش و وزن آن را که برابر $90 و ۱۲ می‌باشد را محاسبه می‌کنیم.
ب) حد آن که$90 است را محاسبه می‌کنیم.
ج) گره غیرامیدبخش است.
۱۷.عقبگرد به گره (۲،۲)
۱۸.گره (۳،٤) را بازدید می‌کنیم.
الف) ارزش و وزن آن را که برابر $40 و ۲ می‌باشد را محاسبه می‌کنیم.
ب) حد آن را که برابر $50 می‌باشد، محاسبه می‌کنیم.

ج) گره امیدبخش نیست.
۱۹.عقبگرد به $(0,0)$
۲۰.گره (۱،۲) را بازدید می‌کنیم.
الف) ارزش و وزن آن را که برابر $0 و 0 می‌باشد را محاسبه می‌کنیم.
ب) حد آن را که برابر $82 می‌باشد، محاسبه می‌کنیم.
ج) گره امیدبخش نیست.
۲۱.عقبگرد به $(0,0)$
الف) ریشه فرزند دیگری ندارد.

دقت کنید در درخت فضای حالت هرس شده فقط ۱۳ گره وجود دارد، در حالی که کل درخت فضای حالت ۳۱ گره دارد.

سود ماکزیمم برابر با ۳۸ و وزن قطعات انتخاب شده برابر ۳۹ خواهد بود.