

1- تابع زمانی زیر مربوط به کدام مسئله بازگشتی می باشد؟

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

الف) فاکتوریل

ب) دنباله فیبوناچی

ج) مرتب سازی ادغامی

د) برج هانوی

پاسخ:

$$n! = \begin{cases} 1 & \text{if } n = 0 \\ n * (n-1)! & \text{if } n > 0 \end{cases} \quad \text{گزینه الف تابع زمانی فاکتوریل (صفحه 39 کتاب)}$$

$$\text{Fib}(n) = \begin{cases} 0 & \text{if } n = 1 \\ 1 & \text{if } n = 2 \\ \text{fib}(n-1) + \text{fib}(n-2) & \text{if } n > 2 \end{cases} \quad \text{گزینه ب تابع زمانی فیبوناچی (صفحه 41 کتاب)}$$

$$T(n) = \begin{cases} d & \text{if } n = 1 \\ 2T\left(\frac{n}{2}\right) + cn & \text{if } n > d \end{cases} \quad \text{گزینه ج تابع زمانی ادغامی (صفحه 28 کتاب)}$$

گزینه صحیح گزینه د (صفحه 25 و 43 کتاب) در این مساله بازگشتی سه محور ثابت به نامهای  $A, B, C$  داریم که در ابتدای کار چند دیسک به اندازه های متفاوت و از بزرگ به کوچک حول محور  $A$  روی هم انباشته شدند هدف آنست که کلیه دیسکها را به میله دیگری مثلا  $C$  انتقال دهیم به صورتی که 1- هر بار بالاترین دیسک باید حرکت داده شود 2- دیسک بزرگتر بر روی دیسک کوچکتر قرار نگیرد 3- در هر بار حرکت فقط یک دیسک را میتوان انتقال داد مراحل انجام کار به صورت زیر است:

1) دو دیسک بالا از سه دیسک محور  $A$  به محور  $B$  منتقل شود

2) آخرین دیسک محور  $A$  به محور  $C$  منتقل شود

3) دو دیسک حول محور  $B$  به محور  $C$  منتقل شود

۳- عملکرد تابع زیر چیست؟

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

الف)  $a^b$

ب)  $b^a$

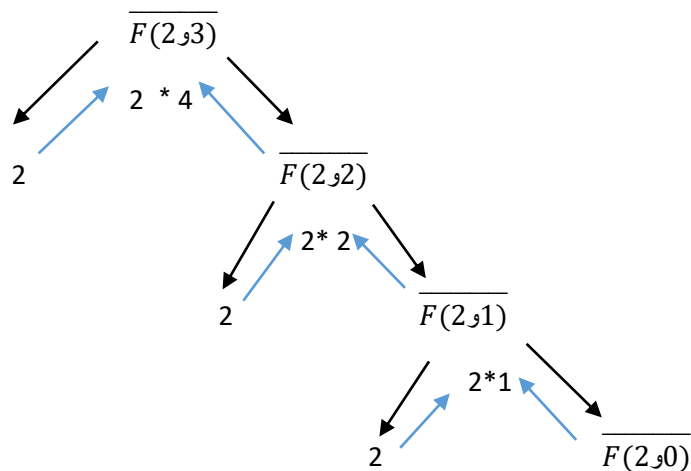
ج)  $a*b$

د)  $a+b$

پاسخ:

برای حل این عملگر ما یک مثال میزنیم و فرایند را تا جایی ادامه میدهیم که  $b=0$  شود

در این مثال  $b=3$  و  $a=2$  میگیریم



جواب برابر شد با  $4*2=8$  گزینه درست الف 2 به توان 3 برابر است با 8

گزینه ب) 3 به توان 2 برابر است با 9

گزینه ج)  $2*3=6$

گزینه د)  $2+3=5$

5- 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)+F(m-1,n-1));
}

```

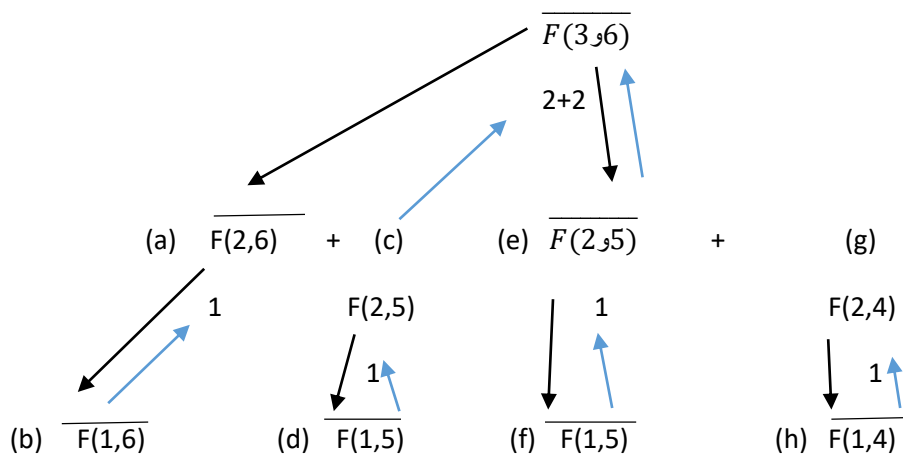
الف) 3

ب) 4

ج) 5

د) 6

پاسخ: مراحل اجرایی الگوریتم بالا را به ازای مقادیر داده شده در شکل زیر نمایش می‌دهیم (صفحه 47 کتاب)



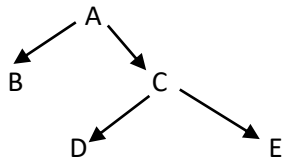
گزینه الف) 3 برابر نیست با 4

گزینه ب) 4 برابر است با 4 گزینه صحیح

گزینه ج) 5 برابر نیست با 4

گزینه د) 6 برابر نیست با 4

7- خروجی تابع بازگشتی زیر برای دودویی شکل زیر چیست؟



الف) 2

ب) 3

ج) 4

د) 5

Int func(Node\*tree)

{

    If (tree==NULL) return0;

    ehse

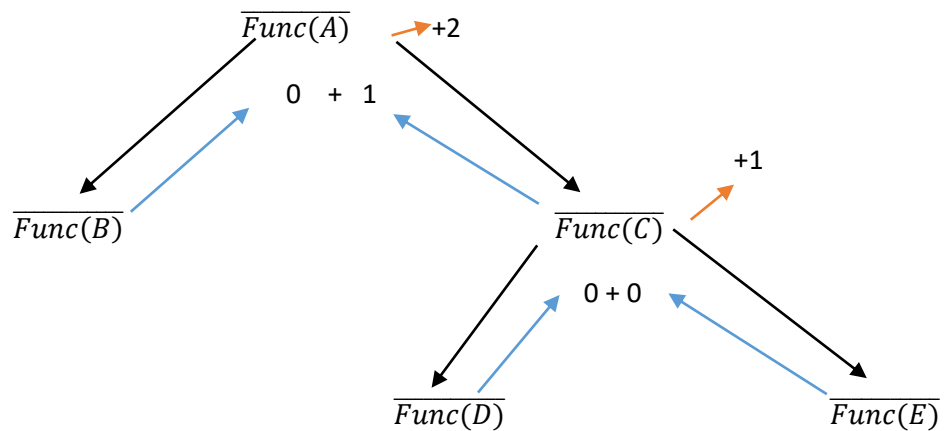
{

        If (tree->left==NULL&&tree->right==NULL) return0;

        Else return(func(tree->left)+func(tree->right)+1);

}

پاسخ: مراحل اجرایی الگوریتم بالا را به ازای درخت داده شده در شکل زیر نمایش می‌دهیم گزینه الف گزینه صحیح است



9-مرتبه زمانی تابع بازگشتی زیر چیست؟

$$T(n)=3T(n/4)+n$$

الف)  $\Theta(n)$

ب)  $\Theta(n^2)$

ج)  $\Theta(n^{\log_4 3})$

د)  $\Theta(n^{\log_3 4})$

پاسخ: (صفحه 61 کتاب) طرف راست رابطه بالا را طریق روش تکرار با جایگذاری، تکرار میکنیم

$$T(n)=3T(n/4)+n$$

$$=3^2T(n/14)+3(n/4)+n$$

=...

$$\leq 3^i T(N/4^i) + n \sum_{j=0}^{i-1} (3/4)^j$$

رابطه بالا را تا زمانی که به  $T(n)$  نرسیدیم ادامه می دهیم بنابراین:

(با فرض اینکه  $n$  توانی از 4 باشد)

$$n/4^i=1 \rightarrow i=\log_4 n$$

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

در عبارت بالا مقدار مجموع برای  $n$  های بزرگ ثابت می باشد، لذا خواهیم داشت :

$$T(n) \leq C_1 n^{\log_4 3} + C_2 n$$

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

$$\lim_{n \rightarrow \infty} \sum_{j=0}^{\log_4 n} \left(\frac{3}{4}\right)^j = 4 \rightarrow C_2 < 4$$

$$n \rightarrow \infty$$

بنابراین در نهایت خواهیم داشت:

$$T(n) \leq C_1 n^{\log_4 3} + 4n$$

$$\rightarrow T(n) \in O(n)$$

گزینه الف درست است

11- اگر جستجو دودویی را بر روی لیست زیر به دنبال عنصر 18 انجام دهیم این عنصر در چندمین مقایسه یافت میشود؟

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

1(د)

2(ج)

3(ب)

4(الف)

پاسخ: صفحه (87 و 88 کتاب) در جستجو دودویی داریم که فرض کنید  $a_i$  بازای  $1 \leq i \leq n$ ، لیستی از عنصر باشد که به صورت صعودی مرتب شده می‌خواهیم وجود یا عدم وجود عنصر 18 مشخص کنیم:

1) لیست  $a$  به دو زیر لیست تقسیم می‌شود (تقریباً با طول برابر)  $\{11, 12, 18, 20, 21\}$  و  $\{27, 40, 75, 80, 85\}$

اگر  $x$  کوچکتر از عنصر میانی باشد زیر لیست چپ انتخاب می‌کنیم در غیر این صورت زیر لیست راست  $21 > 18$   
 $\{11, 12, 18, 20, 21\}$

2) عنصر  $x$  را در لیست جدید در صورتی که لیست به اندازه کافی کوچک باشد جستجو می‌کند. در غیر این صورت عمل تقسیم لیست به دو لیست کوچکتر و دوباره ادامه می‌آید.  $\{11, 12\}$  و  $\{20, 21\}$

عنصر  $18 = 18$  (عنصر در مرحله دوم با میانه برابر و یافت شد) گزینه ج صحیح است

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

1(الف)  $\Theta(n)$

2(ب)  $\Theta(n^2)$

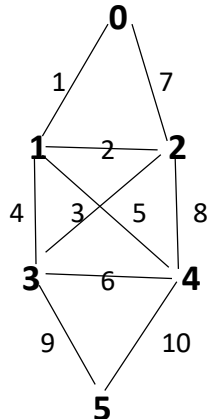
3(ج)  $\Theta(n \log^n)$

4(د)  $\Theta(\log n)$

پاسخ: با توجه به الگوریتم QuickSort procedure اگر آرایه از قبل مرتب باشد  $n$  بار فراخوانی میشود. در واقع فراخوانی اول یک قسمت آرایه شامل صفر عنصر و قسمت دیگر شامل  $n-1$  عنصر خواهد بود یعنی با هر فراخوانی یک عنصر کم میشود. در ضمن هر فراخوانی از مرتبه  $O(n)$  است پس در بدترین حالت (وقتی آرایه مرتب است) مرتبه الگوریتم  $O(n^2)$  خواهد بود. بنابراین این گزینه ب صحیح است.

15- اگر به روش کروسکال درخت پوشا مینیمم را برای گراف شکل زیر بدست آوریم در مرحله سوم کدام یال به درخت اضافه

میشود؟



01(الف)

02(ب)

14(ج)

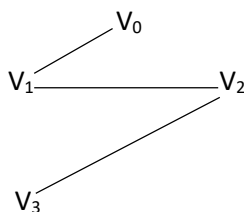
23(د)

پاسخ: (صفحه 151-153 کتاب) نخست درخت heap برای مرتب کردن یال ها بر حسب وزن یال ها تشکیل میشود، که در ریشه درخت  $e_{01}=1$  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  $v=v_1$  می باشد (یال  $e_{01}$  از درخت heap حذف می شود) بنابر این داریم  $F=\{e_{01}\}$

مرحله دوم:  $e_{12}=2$  به عنوان یال کم هزینه انتخاب میشود که در آن  $u=v_1$  و  $v=v_2$  می باشد (یال  $e_{12}$  از درخت heap حذف میشود) بنابر این 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  $v=v_3$  می باشد (یال  $e_{23}$  از درخت حذف میشود) بنابر این داریم:  $F=\{e_{01}, e_{12}, e_{23}\}$



طبق مراحل انجام شده در مرحله سوم یال  $e_{23}$  به درخت

پوشا مینیمم اضافه میشود و گزینه د صحیح است.

17-مساله ضرب زنجیره ای ماتریس ها برای چهار ماتریس با ابعاد زیر مد نظر بگیرید و مشخص کنید ترتیب بهینه ضرب که منجر به کمترین تعداد عمل ضرب میشود کدام گزینه میباشد؟

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

(ج)  $A((BC)D)$

(الف)  $A(B(CD))$

(د)  $(A(BC)D)$

(ب)  $(AB)(CD)$

پاسخ: (صفحه 199 کتاب)

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

(الف)

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

(ب)

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

(ج)

$$(A(BC)D) = 20 \times 2 \times 30 + 20 \times 30 \times 12 + 20 \times 12 \times 8 = 10320$$

(د)

همانطور که ملاحظه میکنید ترتیب گزینه ج برای ضرب این چهار ماتریس ترتیب بهینه است. (ترتیب بهینه فقط به ابعاد ماتریس بستگی دارد)

19- کدام گزینه صحیح نیست؟

الف) اغلب مسایلی که با تکنیک عقب گرد حل میشوند به شکلی هستند که از اصول و مفاهیم درخت ها استفاده میکنند

ب) تکنیک عقب گرد حالت مصطلح شده جستجو عمقی یک درخت میباشد

ج) درخت تصمیم در تکنیک عقب گرد کاربردی ندارد و در سایر روش ها استفاده میشوند

د) در تکنیک عقب گرد چنانچه مساله بیش از یک جواب داشته باشد همه س جواب ها را باید پیدا کرد

پاسخ:

الف) درست (صفحه 251 کتاب)

ب) درست (صفحه 253 کتاب)

ج) نادرست (صفحه 252 کتاب)

د) درست (صفحه 253 کتاب)

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

الف) فضای حالت مساله ای که قرار است با این روش حل شود باید با یک گراف قابل نمایش باشد

ب) این روش شکل بهبود یافته ای از روش تقسیم حل میباشد

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

الف) گزینه الف و ب                      ب) گزینه الف و ب و ج

ج) گزینه الف و ج                      د) فقط گزینه الف

پاسخ:

الف) درست (صفحه 287 کتاب)

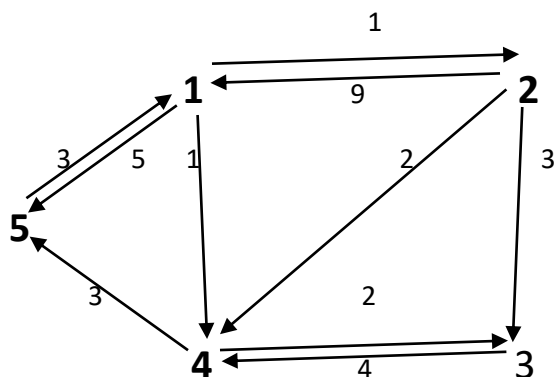
ب) نادرست زیرا این روش شکل بهبود یافته روش عقبگرد است (صفحه 287 کتاب)

ج) درست (صفحه 288 کتاب)

بنابر این گزینه ج صحیح است



23- ماتریس همجواری گراف زیر کدام است؟



$$\begin{bmatrix} 0 & 1 & \infty & 1 & 5 \\ 0 & 0 & 3 & 2 & \infty \\ \infty & \infty & 0 & 4 & \infty \\ \infty & 2 & 2 & 0 & \infty \\ 3 & \infty & \infty & \infty & 0 \end{bmatrix} \text{ (ج)}$$

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

$$\begin{bmatrix} 0 & 1 & \infty & 1 & 2 \\ 0 & 0 & 3 & 2 & \infty \\ \infty & 3 & 0 & 4 & \infty \\ \infty & 2 & 2 & 0 & \infty \\ 5 & \infty & \infty & \infty & 0 \end{bmatrix} \text{ (د)}$$

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

پاسخ:

$$D^{(1)}[2][5] = \min(D^{(0)}[2][1] + D^{(0)}[1][5], D^{(0)}[2][4] + D^{(0)}[4][5], D^{(0)}[2][1] + D^{(0)}[1][4] + D^{(0)}[4][5], D^{(0)}[2][3] + D^{(0)}[3][4] + D^{(0)}[4][5]) = 9+5, 2+3, 9+1+3, 3+4+3 = 5$$

در سه گزینه ب، ج، د  $D^{(0)}[2][5] = \infty$  است پس گزینه الف صحیح است

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

(الف) مساله مرتب سازی یک لیست  $n$  عنصری

(ب) مساله جستجوی آیتام در یک لیست  $n$  عنصری

(ج) مساله رنگ آمیزی گراف

(د) مضرب ضرب دو ماتریس  $n \times n$

پاسخ: (صفحه 305 کتاب) با توجه به اینکه الگوریتم هایی که پیچیدگی زمانی آنها حد اکثر چند جمله ای میباشد از قبیل مرتب سازی ، ضرب ماتریس ها ، درخت های پوشای کمینه و غیره می باشند و الگوریتمهایی که پیچیدگی زمانی آنها نمایی می باشد از قبیل رنگ آمیزی گراف ها ، کوله پشتی 0/1، فروشنده دوره گرد و غیره در عمل کاربرد خاصی ندارند و برای حل یک مثال نه چندان بزرگ ممکن است ماه ها و یا سال ها زمان نیاز داشته باشند گزینه ج صحیح است

تشریحی \*\*\*

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

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

پاسخ: (صفحه 81 کتاب سوال 10)

$T(n)$  را با  $a_n$  نشان میدهیم

$$A_n - 3a_{n-1} - 4a_{n-2} = 0$$

$$X^2 - 3X - 4 = 0$$

$$X = \frac{3 + \sqrt{25}}{2}, x = \frac{3 - \sqrt{25}}{2}$$

$$A_n = C_1 \left( \frac{3 + \sqrt{25}}{2} \right)^n + C_2 \left( \frac{3 - \sqrt{25}}{2} \right)^n = \Theta \left( \left( \frac{3 + \sqrt{25}}{2} \right)^n \right)$$

3- هفت کار طبق جدول جدول زیر با بهره و مهلت معین وجود دارد. با فرض اینکه زمان انجام همه کارها یکسان و برابر با یک واحد زمانی باشد ، با روش حریصانه یک ترتیب بهینه برای انجام کارها به گونه ای ارایه دهید که بیشترین سود حاصل شود؟

بهره	مهلت	کار
60	3	1
50	1	2
30	1	3
20	2	4
15	3	5
10	1	6
5	2	7

پاسخ: (صفحه 176 کتاب)

طبق قضیه 9-5 داریم که خروجی حاصل از الگوریتم زمانبندی با مهلت یک مجموعه بهینه تولید میکند به بیان دیگر سود حاصل از مجموعه امکان پذیر، ماکزیمم می باشد.

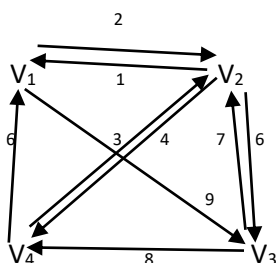
بنابر این بنابر الگوریتم زمانبندی با مهلت نخست مقدار  $z$  را برابر صفر قرار می دهیم.

جدول زیر مراحل اجرایی را نمایش میدهد:

مرحله	J	سود	مجموعه امکان پذیر
0	0	0	هست
1	{1}	60	هست
2	{2,1}	110	هست
3	{2,3,1}	110	نیست
4	{2,4,1}	130	هست
5	{2,4,1,5}	130	نیست
6	{1,2,6,4}	130	نیست
7	{2,4,7,1}	130	نیست

جواب بهینه  $z=\{2,4,1\}$  با سود 130 خواهد بود.

5- گراف شکل زیر را در نظر بگیرید و با استفاده از روش برنامه نویسی پویا یک تور بهینه برای آن بیان کنید؟ (مساله فروشنده دور گرد)



پاسخ: (صفحه 221 کتاب)

باتوجه 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  $V_1$  را در نظر میگیریم و مسافت هر راس تا  $V_1$  را پیدا میکنیم. با توجه به ماتریس مجاورتی از  $V_3$  به  $V_1$  یالی وجود ندارد از علامت  $\infty$  استفاده میکنیم. حال همه مجموعه های حاوی یک عنصر را در نظر میگیریم:

$$D[V_3][\{V_2\}] = \min_{j \in \{V_2\}} (W[3][j] + D[V_1][\{V_2\} - \{V_j\}])$$

$$V_j \in \{V_2\} \Rightarrow W[3][2] + D[V_1][0] = 7 + 1 = 8$$

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 مجموعه های جاوی دو عنصر را در نظر میگیریم:

$$D[V_4][\{V_2, V_3\}] = \min_{V_j \in \{V_2, V_3\}} (W[4][j] + D[V_j][\{V_2, V_3\} - \{V_j\}])$$

$$V_j \in \{V_2, V_3\}$$

$$= \min_{V_j \in \{V_2, V_3\}} (W[4][2] + D[V_2][\{V_3\}], W[4][3] + D[V_3][\{V_2\}])$$

$$\min_{V_j \in \{V_2, V_3\}} = (3 + \infty, \infty + 8) = \infty$$

به طور مشابه:

$$D[V_3][\{V_2, V_4\}] = \min_{V_j \in \{V_2, V_4\}} (W[3][j] + D[V_j][\{V_2, V_4\} - \{V_j\}])$$

$$D[V_2][\{V_3, V_4\}] = \min_{V_j \in \{V_3, V_4\}} (W[2][j] + D[V_j][\{V_3, V_4\} - \{V_j\}])$$

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

$$D[V_1][\{V_2, V_3, V_4\}] = \min_{V_j \in \{V_2, V_3, V_4\}} (W[1][j] + D[V_j][\{V_2, V_3, V_4\} - \{V_j\}])$$

$$V_j \in \{V_2, V_3, V_4\}$$

$$\min_{V_j \in \{V_2, V_3, V_4\}} = (W[1][2] + D[V_2][\{V_3, V_4\}], W[1][3] + D[V_3][\{V_2, V_4\}], W[1][4] + D[V_4][\{V_2, V_3\}])$$

$$\min_{V_j \in \{V_2, V_3, V_4\}} = (2 + 20, 9 + 12, \infty + \infty) = 21$$

\*\*\*سوالات زوج نیم سال دوم 95-96

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

A x=0;

For (i=0; i<n; i++)

For (j=1 ; j<=[i]; j++)

X++;

الف)  $O(m)$

ب)  $O(n)$

ج)  $O(nm)$

د)  $O(n+m)$

پاسخ: (مشابه سوال 2 صفحه 32 کتاب)

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

پس هزینه کل برابر است با :

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

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

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

بنابر این گزینه د صحیح می باشد.

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

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 0 \quad (\text{الف})$$

$$f(n) \in O(g(n)) \quad \text{و} \quad g(n) \in O(f(n)) \quad (\text{ب})$$

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 10^5 \quad (\text{ج})$$

$$h(n) \in g(n) \quad \text{و} \quad f(n) \in O(h(n)) \quad (\text{د})$$

پاسخ: (صفحه 14 کتاب)

الف) طبق قضیه 1-4 از این رابطه نتیجه میشود  $T(n) \in O(g(n))$  و  $T(n) \notin \Theta(g(n))$  پس این گزینه نادرست است

ب) طبق قضیه 1-3 این رابطه نادرست است زیرا داریم:  $g(n) \in \Omega(f(n))$  اگر و فقط اگر  $f(n) \in \Theta(g(n))$

ج) طبق قضیه 1-4 از این رابطه نتیجه میشود که  $f(n) \in \Theta(g(n))$  پس این گزینه صحیح میباشد

د) طبق قضیه 1-3 این رابطه نادرست است زیرا داریم:  $T(n) \in \Theta(h(n))$  و  $g(n) \in \Theta(h(n)) \rightarrow T(n) \in \Theta(h(n))$

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

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

الف) 25

Int f(int a[], int n){

ب) 30

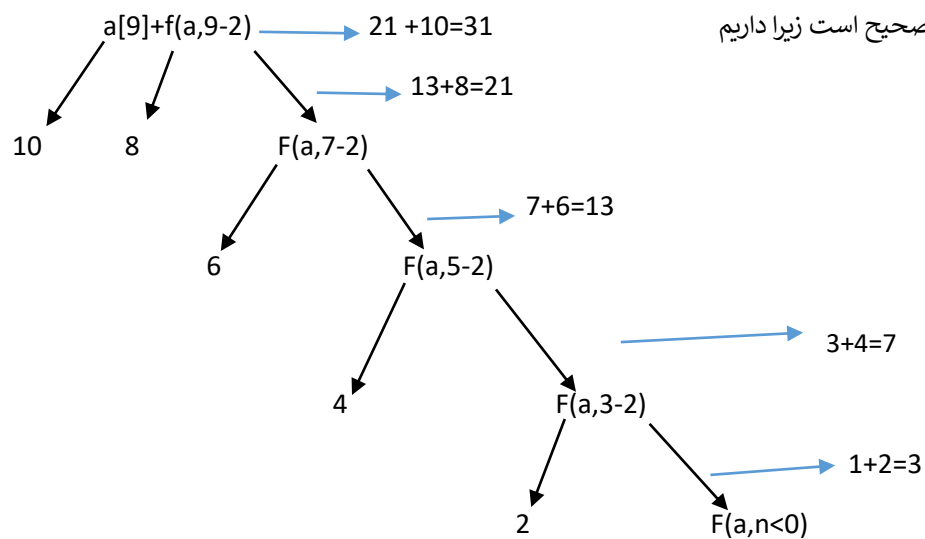
If (n<0) return 1;

ج) 31

Return a[n]+f(a,n-2);

د) 55

}



8- زمان اجرایی تابع بازگشتی زیر کدام است؟

Int f(int n){

الف)  $O(n^2)$

If(n<=2) return 1;

ب)  $O(2^n)$

Return f(n-1)\*f(n-2);

ج)  $O(n \log n)$

}

د)  $O(2^{n/2})$

پاسخ: (تابع فیبوناچی است)

$$T(n) > 2 * T(n-2)$$

$$> 2 * 2 * T(n-4)$$

$$> 2 * 2 * 2 * T(n-6)$$

...

$$> 2 * 2 * 2 * \dots * 2 * T(0)$$

$$\xleftarrow{\frac{n}{2} \text{ مرتبه}}$$

باتوجه 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 صدا 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 نیاز دارد:

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

حکم استقرا: ثابت میکنیم به ازای  $n$  نیز درست می باشد 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.

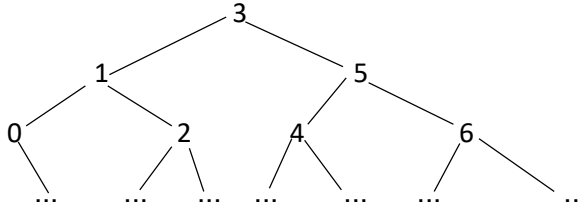
گزینه د صحیح است.

10- در الگوریتم جستجو دودویی متوسط تعداد مقایسه ها در جستجو ناموفق برای یک آرایه 7 عنصری چقدر است؟

الف)  $\frac{17}{7}$       ب) 3      ج)  $\frac{17}{8}$       د) 4

پاسخ: (صفحه 90 کتاب)

با توجه به درخت جستجوی دودویی داریم:



$= (1+7*3)/8 = 3$  متوسط تعداد مقایسه در جستجوی ناموفق

گزینه ب صحیح است.

12- 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

(الف) برنامه نویسی پویا = 512 استراسن = 343

(ب) برنامه نویسی پویا = 64 استراسن = 49

(ج) برنامه نویسی پویا = 512 استراسن = 392

(د) برنامه نویسی پویا = 64 استراسن = 392

پاسخ: (صفحه 199 و صفحه 107 کتاب)

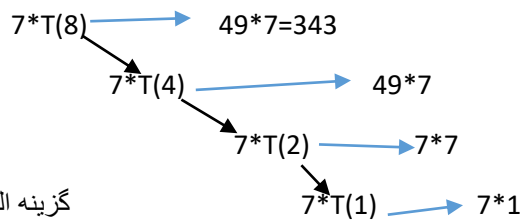
به طور کلی برای ضرب یک ماتریس  $i*j$  در یک ماتریس  $j*k$  با استفاده از ضرب استاندارد ماتریس ها در برنامه نویسی پویا تعداد اعمال ضرب لازم  $i*j*k$  خواهد بود یعنی داریم:  $8*8*8=512$

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

بنابر این تعداد عمل ضرب برابر است با:

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

پس داریم :



گزینه الف صحیح است.



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

کار	1	2	3	4	5	6	7	8	9	10	11	12
مهلت	1	7	6	4	2	6	4	3	2	4	7	5
سود	19	7	25	15	6	16	27	10	3	23	4	2

الف) 2 و 11 و 6 و 10 و 3 و 7 و 1      ب) 11 و 2 و 6 و 10 و 7 و 3 و 1      ج) 11 و 2 و 6 و 10 و 3 و 7 و 1      د) 2 و 11 و 6 و 10 و 7 و 3 و 1

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 هارا بر اساس اولویت سود به صورت غیر نزولی مرتب کرده وبا استفاده از روش حریصانه انتخاب میشود که چه کاری را باید زود تر انجام داد.

کار	12	9	11	5	2	8	4	6	1	10	3	7
سود	2	3	4	6	7	10	15	16	19	23	25	27

گزینه ب و د نادرست است زیرا سود کار 7 بیشتر از کار 3 است

از میان گزینه الف و ج گزینه ج صحیح است زیرا سود کار 2 بیشتر از کار 11 است .

16- برای ضریب دو جمله ای با استفاده از الگوریتم زیر چه تعداد عمل جمع مورد نیاز است؟

```

Int b(int n , int k){
    If( k==0 | n==k) return 1;
    Return b(n-1 , k-1)+b(n-1 , k);
}

```

د)  $\frac{k(2n-k-1)}{2}$

ج)  $2\binom{n}{k} - 1$

ب)  $2\binom{n}{k} - 2$

الف)  $\binom{n}{k} - 1$

پاسخ: (صفحه 204 کتاب)

این الگوریتم محاسبه ضرب دو جمله ای با استفاده از تقسیم و حل می باشد. هر عامل باید آن قدر بسط داده شود که در نهایت به 1 برسد و یک هارا باید با هم جمع کرد. قطعاً باید  $\binom{n}{k}$  عدد را با هم جمع کنیم که  $\binom{n}{k}$  شود، در این صورت تعداد اعمال جمع  $1 - \binom{n}{k}$  خواهد بود.

گزینه الف صحیح است.

18- فرض کنید قرار است بین دو تیم A و B یک مسابقه انجام شود. اولین تیمی که تعداد بردش n شود برنده مسابقه است. با فرض اینکه بازی نتیجه مساوی نداشته باشد حد اکثر تعداد بازی هایی که باید برگزار شود چقدر است؟

الف) n      ب) n+1      ج) 2n-1      د) 2n

پاسخ: (صفحه 242 کتاب)

چون ممکن است یک در میان، تیمی بر تیم دیگر غلبه کند وقتی که اولین تیم به n برد میرسد تیم دوم n-1 برد خواهد داشت پس حد اکثر تعداد مسابقه ایی که برگزار میشود 2n-1 بازی خواهد بود.

بنابر این گزینه ج صحیح است.

20- تعداد درخت های جستجو دودویی با n کلید متمایز کدام است؟

الف)  $\frac{1}{n+1} \binom{2(n-1)}{n-1}$       ب)  $\frac{1}{n+1} \binom{2n}{n}$       ج)  $\frac{1}{n} \binom{2(n-1)}{n-1}$       د)  $\frac{1}{n} \binom{2n}{n}$

پاسخ: (صفحه 248 کتاب سوال 16)

چون n راس و هر راس دو فرزند میتواند داشته باشد 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 می توان کلید ها 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:  $\frac{1}{n+1} \binom{2n}{n}$

گزینه ب صحیح است.

22- کدام گزینه درست است؟

الف) روش عقبگرد میتواند مرتبه زمانی مسایل سخت را کاهش دهد.

ب) روش عقبگرد اصلاح شده جستجو ردیفی یک درخت است.

ج) مسایلی که به روش عقبگرد حل میشوند اغلب مسایل تصمیم گیری هستند.

د) در روش عقبگرد تنها بهینه ترین جواب از میان همه جواب های مساله بدست می آید.

پاسخ: (از صفحه 252 تا صفحه 253 کتاب)

الف) نادرست زیرا روش عقبگرد نمیتواند مرتبه زمانی مسایل سخت را کاهش دهد و فقط با کاهش حالت ها تنها زمان اجرا برای طول داده (n) کوچک کاهش میدهد.

ب) نادرست زیرا روش عقبگرد اصلاح شده روش عمقی است.

ج) درست است.

د) نادرست زیرا در روش عقبگرد چنانچه مساله بیش از یک جواب داشته باشد همه جواب ها را باید پیدا کنیم.

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

الف)  $\frac{n(n-1)}{n+1}$       ب)  $(n-1)^n - 1/n-2$       ج)  $(n-1)^n + 1/n+1$       د)  $(n-1)^{2n} + 1/n-2$

پاسخ: (صفحه 272 کتاب)

تعداد گره های درخت فضای حالت برای این الگوریتم عبارت است از:

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

بنابر این گزینه ب صحیح است.

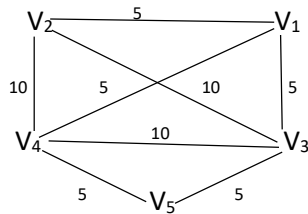
تشریحی \*\*\*

2-یک صفحه مدار چاپی دارای 5حفره 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 1ام را به حفره 2ام 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: (صفحه 150 کتاب)

با استفاده از گراف و الگوریتم کروسکال داریم:



گراف ماتریس به شکل فوق میباشد با استفاده از روش کروسکال ابتدا کوچک ترین یال را که در این جا از میان یالهای  $\{E_{12}, E_{13}\}$  یکی را انتخاب کرده به طور مثال  $E_{13}$  را انتخاب کرده سپس از  $V_3$  به بقیه راس ها یال کوچک تر را که  $E_{35}$  است و پس از آن از یال  $E_{54}$  به  $V_4$  رفته و بعد از آن چون هر یال از  $V_4$  به  $V_2$  ایجاد دور میکند ما کوچک ترین 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:  $5+5+5+5=20$

4- با استفاده از روش عقبگرد برای مساله کوله پشتی صفر و یک سود ماکزیمم از 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: (مشابه مثال صفحه 274 کتاب)

سود ماکزیمم برابر با 38 و وزن قطعات انتخاب شده برابر 39 خواهد بود.