طراحی الگوریتم

نمونه سوالات امتحانی

تابستان ۹۹

بسم تعالی

نام و نام خانوادگی : فاطمه طیبی
شماره دانش جویی : ۹۷۰۱۴۷۰۷۷
دانشگاه پیام نور واحد تهران شمال

نمونه سوالات شامل :
سوالات زوج نیمسال اول ۹۷-۹۸
سوالات فرد نیمسال اول ۹۴-۹۵

تستی

۱- پیچیدگی قطعه کد زیر کدام است ؟

```
for ( i=1 ; i<=n ; i++ )
        for ( j=i ; j<= min ( i , k ) ; j++ )
                x++;
```

۱. $O(n^2)$ ۲. $O(1)$ ۳. $O(n)$ ۴. $O(n^{0.5})$

این سوال را برای k=4 و n=6 انجام میدهیم ، نتیجه را بررسی می کنیم . مقدار تکرار طبق جدول زیر بدست می آید.

تعداد تکرار	تغییرات j	i
۱	1	1
۱	2	2
۱	3	3
۱	4	4
0	-	5
0	-	6

بدین صورت مشخصات که هر بار شارش i ، اندیس j هم حداکثر یک بار شارش می کند و بیشترین مرتبه زمانی قطعه کد ، زمانی است که n=k باشد که آنگاه به تعداد n بار دستور x++ اجرا میشود. در غیر این صورت حتی کمتر از n دفعه اجرا میشود.

پیچیدگی زمانی از مرتبه O(n) میباشد پس گزینه ۳ صحیح است.

۲- اگر و فقط اگر ثابت C و ثابت صحیح n_0 ای وجود داشته باشد که برای همه مقادیر $n \geq n_0$ داشته باشیم :

$T(n) \leq Cg(n)$ ، آنگاه میتوانیم بگوییم : ....

۱. $T(n) \in O(g(n))$ ۲. $T(n) \in \theta(g(n))$

۳. $T(n) \in \Omega(g(n))$ ۴. $T(n) \in \Delta(g(n))$

این تعریف یعنی به ازای تمام n های بزرگتر از n_0 همواره رشد g(n) یا بیشتر از T(n) است یا مساوی یعنی T(n) کران پایینی برای g(n) محسوب میشود و رشد T(n) حداکثر به

اندازه g(n) میتواند باشد و هرگز بزرگ تر از آن نخواهد شد. بدین صورت نوشته میشود $T(n) \in O(g(n))$: بنابراین گزینه یک صحیح است.

۳- کدام یک از روابط زیر در مورد پیچیدگی زمانی یک الگوریتم صحیح نیست ؟

۱. $\text{if} \begin{bmatrix} \lim \frac{T(n)}{g(n)} = 0 \\ n \to \infty \end{bmatrix} \Rightarrow T(n) \in O(g(n))$

۲. $\text{if} \begin{bmatrix} \lim \frac{T(n)}{g(n)} = +\infty \\ n \to \infty \end{bmatrix} \Rightarrow g(n) \in O\big(T(n)\big),\ T(n) \notin \theta(g(n))$

3. $\text{if } (T(n) = \theta\big(g(n)\big)) \Leftrightarrow g(n) \in \theta(T(n))$

۴. $\text{if } (T(n) = o\big(g(n)\big) \Leftrightarrow T(n) \in \Omega(g(n))), (T(n) = O\big(g(n)\big)$

گزینه ۱ : این حد یعنی با رشد n به سمت بینهایت کسر $\frac{T(n)}{g(n)}$ به سمت صفر میل میکند یعنی همواره g نسبت به T بزرگ و بزرگتر میشود . پس هرگز T بزرگتر از g نمیشود یعنی نهایتا به اندازه g میشود.

با توجه به $T(n) \in O\big(g(n)\big)$ پسگزینه یک درست است.

گزینه ۲ : این حد یعنی با رشد n به سمت T همواره بزرگتر از g میشود پس عبارت $T(n) \in O\big(g(n)\big)$ درست است یعنی g حداکثر به اندازه T میتواند رشد کند نه بیشتر و عبارت $T(n) \notin \theta(g(n))$ هم درست است چون T به اندازه میانگین g نیست پس گزینه ۲ درست است.

گزینه ۳ : بیان میکند که اگر T به اندازه میانگین g باشد در نتیجه g به اندازه میانگین T رشد دارد که شرط "$\in \theta$" میتواند یک رابطه دو شرطی باشد که این گزینه هم درست است.

گزینه ۴ : رابطه دو شرطی $T(n) \in \Omega(g(n)) \Leftrightarrow T(n) = o\big(g(n)\big)$ یک رابطه اشتباه است و T نمیتواند حداکثر اندازه g باشد و هم حداقل ، رشد تابع T هم درست است حداکثر به اندازه g عنوان شد پس گزینه ۴ نادرست است.

۴- مرتبه اجرایی رابطه بازگشتی زیر ، برابر کدام گزینه است ؟

$$\begin{cases} T(1) = 1 \\ T(n) = 3T\left(\frac{n}{4}\right) + n \operatorname{Log} n \end{cases}$$

۱. $\theta(n^2 \operatorname{Log} n)$ ۲. $\theta(n^2)$ ۳. $\theta(n^{\frac{3}{4}})$ ۴. $\theta(n \operatorname{Log} n)$

جواب :

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

$$n^{\operatorname{Log}_b^a} = n^{\operatorname{Log}_4^3} = n^{0.7}, O(f(n)) = n \operatorname{Log} n > n^{0.7}$$

$$\Rightarrow T(n) = \theta(n \operatorname{Log} n)$$ گزینه ۴ صحیح میباشد

۵- تابع بازگشتی زیر بر روی درخت دودویی T چه عملی را اننجام میدهد ؟

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

۱. تعداد برگ های درخت را میشمارد.   ۲. ارتفاع درخت را محاسبه میکند.

۳. تعداد کل گره های درخت را میشمارد.   ۴. تعداد گره های داخلی درخت را میشمارد.

جواب : تابع ابتدا بررسی میکند با if اول که اگر گره وجود دارد پس با if دوم بررسی میکند که اگر گره مورد نظر فرزند چپ و راست ندارد عدد یک رابرگرداند یعنی تا اینجا فعلا گره های برگ را شمارش میکند .

دستور else هم یعنی در صورتی که گره مورد نظر فرزند چپ وراست داشته باشد تعداد آنها را با ۱ (که خود گره مورد بررسی است) جمع کن.

در کل این تابع بازگشتی تعداد کل گره های درخت را شمارش میکند بنابراین گزینه ۳ درست است.

۶- یافتن بزرگترین عنصر در یک لیست مرتب، از چه مرتبه زمانی ای است ؟ (الگوریتم بهینه)

۱. $\theta(n)$ ۲. $\theta(1)$ ۳. $\theta(n^2)$ ۴.$\theta(n \text{ Log } n)$

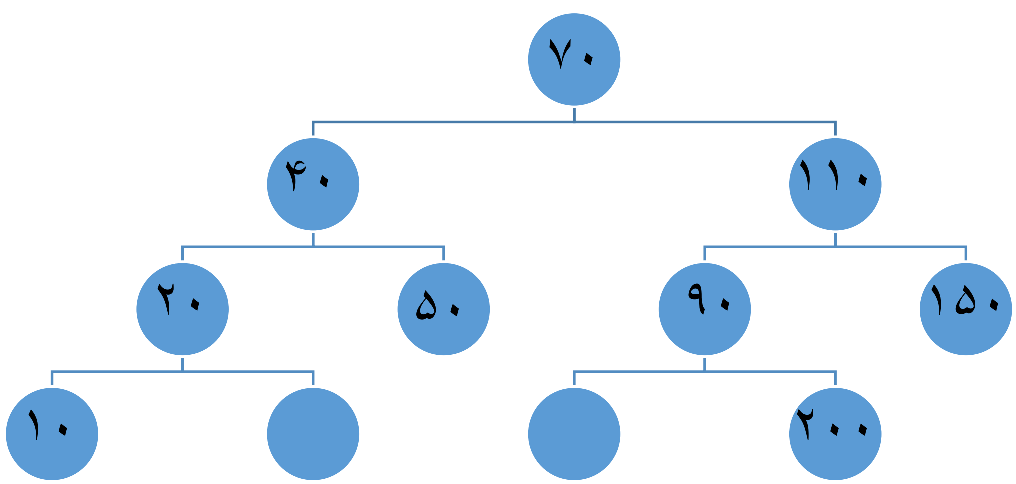
جواب : وقتی از قبل بدانیم که داده های یک لیست مرتب شده اند، برای پیدا کردن داده ماکزیمم کافی است به منتهی الیه آدرس داده ها که حاوی ماکزیمم میباشد مراجعه کنیم و فقط یک عمل مقایسه انجام میشود در نتیجه از مرتبه زمانی O(1) میباشد. مرتبه زمانی عنصر مینیمم نیز مشابه ماکزیمم است و از نوع O(1) میباشد.

۷- میانگین تعداد مقایسه ها برای جستجوی موفق در الگوریتم جستجوی دودویی برای آرایه زیر کدام است ؟

X[0]	X[1]	X[2]	X[3]	X[4]	X[5]	X[6]	X[7]	X[8]
10	20	40	50	70	90	110	150	200

۱. $\frac{23}{9}$ ۲. $\frac{25}{9}$ ۳.$\frac{29}{9}$ ۴. $\frac{31}{3}$

جواب : درخت جستجوی دودویی را تشکیل میدهیم.



طبق درخت فوق مشاهده میکنیم که تعداد مقایسه برای جستجو موفق داده ها به صورت ذیل می باشد :

جست وجوی داده	مقایسه داده
۷۰	۱
۱۱۰	۲
۹۰	۳
۱۵۰	۳
۲۰۰	۴
۴۰	۲
۵۰	۳
۲۰	۳
۱۰	۴

در نتیجه میانگین تعداد مقایسه ها :

$$\frac{1+2+3+3+4+2+3+3+4}{9} = \frac{25}{9}$$ گزینه 2

۸- خاصیت « بهینه زیرساختاری » به چه معنی است ؟

۱. به این معنی است که یک مسئله دارای جواب بهینه است هرگاه هر زیر مسئله آن دارای جواب بهینه باشد.
۲. به این معنی است که یک مسئله دارای جواب بهینه است هرگاه حداقل یک زیر مسئله آن دارای جواب بهینه باشد.
۳. به این معنی است که یک مسئله دارای جواب بهینه است هرگاه حداکثر یک زیر مسئله آن دارای جواب بهینه باشد.
۴. به این معنی است که یک مسئله دارای جواب بهینه است هرگاه هیچ یک زیر مسئله آن دارای جواب بهینه نباشد.

جواب : گزینه یک

۹- پیچیدگی زمانی الگوریتم مرتب سازی سریع (Quick Sort) در بهترین حالت ....، در حالت متوسط .... و در بدترین حالت .... است. (به ترتیب ازراست به چپ)

۱. $\theta(n^2)\,,\theta(n\log n)\,,\theta(n\log n)$
۲. $O(n^2)\,,O(n\log n)\,,O(l)$
۳. $\theta(n^2)\,,\theta(n\log n)\,,\theta(\log n)$
۴. $\theta(n^2)\,,\theta(n\log n)\,,\theta(n)$

جواب : در الگوریتم سریع در بدترین حالت T(n) در حالت کلی به صورت زیر میباشد :

$$T(n) = \begin{cases} 0 & n < 1 \\ T(n-1) + n - 1 & n \geq 1 \end{cases}$$

در نتیجه مجموع جملات بازگشتی به صورت $T(n) = \frac{n(n-1)}{2}$ خواهد بود بنابراین از مرتبه زمانی $T(n) \in \theta(n^2)$ میباشد.

در حالت متوسط رابطه بازگشتی زیر برقرار است :

$$a_n = \begin{cases} 0 & n < 1 \\ a_{n-1} + \dfrac{2(n-1)}{n(n+1)} & n \geq 1 \end{cases}$$

در نتیجه برای n های بزرگ :

$$a_{n-1} + \frac{2(n-1)}{n(n+1)} = a_{n-1} + 2\left(\frac{2n-1}{n} - \frac{2n}{n+1}\right)$$

که پساز حل با روش تکرار و جایگزینی در نهایت خواهیم داشت :

$$a_n \leq 2 \log nn$$

در نتیجه :

$$\frac{T(n)}{n+1} \leq 2 \log nn \Rightarrow T(n) \leq 2(n+1) \log nn$$

بنابراین پیچیدگی زمانی برابر است با :

$$T(n) \in \theta(n \log n)$$

و بهترین حالت الگوریتم مرتب سازی سریع زمانی است که عنصر محور همواره وسط آرایه قرار دارد. آنگاه پیچیدگی زمانی برابر است با :

$$T(n) = 2T\left(\frac{n}{2}\right) + \theta(n) \Rightarrow T(n) = \theta(n \log n)$$

در نتیجه گزینه یک شامل موارد بدست آمده میباشد.

۱۰- پیچیدگی زمانی الگوریتم دیکسترا از چه مرتبه ای است ؟

۱. O(n)　　۲. $O(n^3)$　　۳. $O(n^2)$　　۴. O(n Log n)

جواب : در الگوریتم دیکسترا درهر بار ، فاصله هر گره با گره های قبلی مقایسه میشود. در نتیجه مرتبه زمانی آن $\theta(n^2)$ میباشد . n بیانگر مقدار رئوس میباشد. در نتیجه گزینه سه درست است.

۱۱- تعداد مقایسه ها در الگوریتم بازگشتی پیدا کردن ماکزیمم و مینیمم عنصر در (یک آرایه) به روش تقسیم و غلبه در بدترین حالت کدام است ؟

۱. 2(n-1)　　۲. (n-1)　　۳. $\frac{n}{2}-1$　　۴. $\frac{3n}{2}-2$

جواب : الگوریتم مذکور مطابق زیراست.

```
if (s[1] < s[2] 0 { min = s[1 ; max = s[2]; }
else { min = s[2]; max = s[1]; }
for ( i=3 ; i ≤ n-1 ; i=i+2 ) {
    if ( s[i] > s[i+1] )
        swap ( s[i] , s[i+1] );
if ( s[i] < min ) min = s[i];
if ( s[i+1] > max ) max = s[i+1];
}
```

در این الگوریتم حلقه for برای nهای زوج $(\frac{n}{2}-1)$ بار اجرا میشود و در هربار ۳ مقایسه صورت میگیرد و یک مقایسه هم در ابتدای کار و بیرون حلقه انجام میشود .

گزینه ۴ $\Leftarrow$ $T(n) = 1+\left(\frac{n}{2}-1\right).3 \longrightarrow T(n) = \frac{3n}{2}-2$

و برای n های فرد : $T(n) = \frac{3n}{2}-\frac{3}{2}$

۱۲- کمترین زمان انتظار برای $P_1, P_2, \dots, P_n$ زمانی حاصل میشود که ....

۱. آنها را به ترتیب غیرصعودی بر حسب زمان ارائه خدماتشان مرتب کنیم.

۲.آنها را به ترتیب غیر نزولی برحسب زمان ارائه خدماتشان مرتب کنیم.

۳. آنهارا به ترتیب ورودشان به صف ، سرویس دهی کنیم. (FIFO)
۴. در هر ترتیبی، کمترین زمان انتظار حاصل خواهد شد.

جواب : زمان انتظار هنگامی کمینه میشود که کارها بر مبنای افزایش زمان ارائه خدمات مرتب شوند.یعنی ابتدا به کوتاه ترین کارها سرویس دهی شود. جمع زمانی برگشت (یا جمع کل سیستم) حداقل خواهد بود.

الگوریتم زمان بندی بر مبنای کنیه از مرتبه زمانی $\theta(n \log n)$ میباشد.

بنابراین گزینه ۲ صحیح است.

13- پیچیدگی زمانی الگوریتم کورسکال در بدترین حالت کدام است؟

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

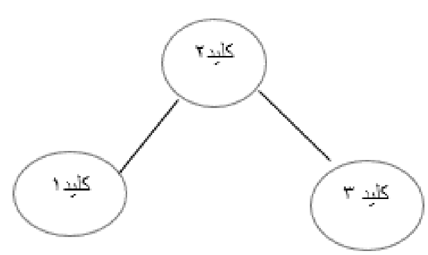
جواب : بدترین حالت الگوریتم کروسکال زمانی است که گراف کامل است. در نتیجه خواهیم داشت :

$$m=|E|=\frac{n(n-1)}{2} \gg T(m) \epsilon \Theta(m \log m) = \theta(n^2 \times 2 \log n) = \Theta(n^2 \log n)$$

گزینهٔ ۴

۱۴- با داشتن احتمالات مربوط به جستجوی کلید مورد جستجو در یک درخت دو دویی جستجو، طبق جدول زیر، زمان جستجوی میانگین برای درخت جستجوی زیر بابر خواهد بود با : ......

کلید	۱	۲	۳
احتمال	۰/۷	./۲	۰/۱



جواب :

	۰	۱	۲	۳
۱	۰	۱	۱	۱
۲		۰	۲	۲
۳			۰	۳
٤				۰

R

	۰	۱	۲	۳
۱	۰	۰/۷	۱/۱	۱/٤
۲		۰	۰/۲	۰/٤
۳			۰	۰/۱
٤				۰

قطر

قطر

قطر

قطر ۱: از آنجا که در قطر ۱ داریم $A[I][J]=P_I$ احتمالات داده شده را در قطر مربوطه قرار می دهیم. به همین ترتیب: R[I][J]=I

قطر ۲:

$A[1][2] = \min (\underline{0+0\ /2}\ ,\underline{0/7+0}\ +(0/7+0/2)=1/1$

K=1 k=2

پس در خانه R[1][2] عدد K=1 را قرار می دهیم.

$A[2][3]=\min(\underline{\ 0 + 0/1}, \underline{\ 0/2 + 0)} + 0/2 + 0/1 =0/4$

K=2 k=3

و در خانه R[2][3] عدد K=2 را قرار می دهیم.

قطر ۳:

$A[1][3]=\min(\underline{0+0/4}\ ,\underline{0/7+0/1}\ ,\underline{1/1+0})+ 0/7+0/2+0/1=1/4$

K=1 K=2 K=3

و در خانه R[1][3] عدد K=1 قرار می دهیم. پس درخت جستجوی دوایی بهینه با کلیدهای KEY_1 تا KEY_3 در کل به زمان جستجوی میانگین ۱/۴ نیاز دارد. گزینۀ ۴

۱۵- اشیاء زیر را در نظر بگیرید، اگر ظرفیت کوله پشتی ۴۰ باشد جواب بهینه برای این کوله پشتی با استفاده از روش حریصانه کدام است؟

X_5	X_4	X_3	X2	X_1	X_i
20	10	15	5	8	P_I
15	8	25	15	16	W_I

39.9-1 41.1-2 38.9-3 40.1-4

جواب : اشیا را بر حسب بیشترین نسبت ارزش بر وزن مرتب می کنیم.

به ترتیب الویت ها اشیا را انتخاب

می کنیم تا زمانی که مجموع وزن اشیا

انتخاب شده بیشتر از ظرفیت کوله پشتی

یعنی ۴۰ نباشد.

انتخاب ها = $X_5 + X_4 + X_3 + X_1 + X_2$

↓ ↓ ↓ ↓ ↓

وزن ۱۵ ۸ ۲۵ ۱۶ ۱۵

تا اینجا مجموع وزن ها برابر است با ۳۹ می بینیم که هنوز ۱ کیلو ظرفیت مانده ولی با انتخاب هر کدام از اشیاء باقیمانده مجموع وزن ها از ظرفیت کوله پشتی بیشتر می شود. در نتیجه تنها راه این است که به اندازۀ یک کیلو (فرضاً واحد وزن نوشته شده کیلو باشد) از بالاترین شیء باقیمانده بر میداریم که شیء X_3میباشد. که یک کیلو از آن به ارزش $\frac{15}{25}$ می باشد. $\frac{15}{25}$= ۰/۶ ←

مجموع انتخاب ای بهینه= تو هیچ کدام از گزینه ها نیست ← ۳۹/۶ = ۰/۶ + ۳۹

۱۶-فرض کنید برای n=7 کارها مهلت و بهره های مربوط به کار ها را به صورت زیر داریم جواب بهینه با الگوریتم زمانبندی با مهلت کدام است؟

بهره	**مهلت**	**کار**

1	3	60
2	1	50
3	1	30
4	2	20
5	3	15
6	1	10

۱-جواب بهینه {۱،۲،۶،۴ } با سود دهی ۱۳۰ خواهد بود.

۲-جواب بهینه {۲،۴،۱،۵} با سود دهی ۱۳۰ خواهد بود.

۳-جواب بهینه {۲،۴،۱} با سود ۱۳۰خواهد بود.

۴- جواب بهینه {۲،۴،۷،۱} با سود ۱۳۰ خواهد بود.

جواب : در مسألۀ رنگ آمیزی گراف در ابتدای کار گره اول هر کدام از m رنگ را می تواند استفاده کند پس با انشعابات خارج شده از گره اول سراغ گره دوم می رویم که تمام m رنگ را جلوی آن درج می کنیم و رنگی که محدودیت استفاده را دارد با ضربدر به عنوان حالت غیر امید بخش مشخص می کنیم و به همین ترتیب الی آخر . در نتیجه در هر بار تمام رنگ های مسأله درج می شود و مجاز و غیر مجاز ها روی آن مشخص می شود مثال مسأله زیر برای ۴ گره و ۳ رنگ می بینیم که حد اکثر انشعاب خارج شده از هر گره نشان دهندۀ تمام رنگ های مسئله است یعنی گزینۀ ۱

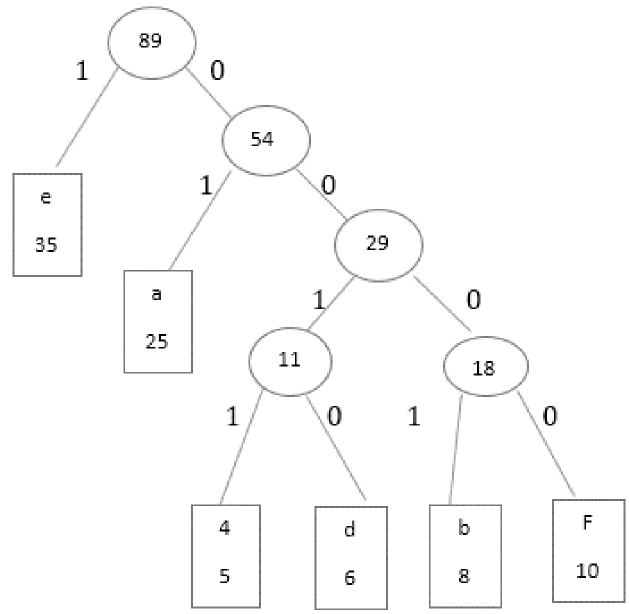
۱۷- فرض کنید متنی شامل حروف a, b, c, d, e, f باشد تعداد کاراکتر های این متن بابر با ۸۹ بیت است که در آن تعداد تکرار کاراکتر ها بصورت زیر آمده است: تعداد بیت های لازم برای ذخیره سازی این متن کدام است؟

کاراکترها	a	b	c	d	e	f
تعداد کاراکترها	25	8	5	6	35	10

۱-۱۷۷ ۲-۲۵۶ ۳-۵۶۷ ۴-۷۱۲

سؤال ۱۷ تستی: در حالت عادی چون هر کار کند ۱ بایت یعنی ۸ بیت فضا نیاز دارد»۸۹ کاراکتر به تعداد ۷۱۲= ۸×۸۹ بیت خانه از حافظه نیاز دارد.

اما با روش کد گذاری هافمن



تعداد تکرار		کد بیتهای مورد نیاز
$2 \times 25 = 50$ ⇨	=>	a = 0 1
$4 \times 8 = 32$	=>	b = 0001
$4 \times 5 = 20$	=>	$c = 0011$
4× 6 =24	=>	d=0010
1× 35=35	=>	e=1
4×10= 40	=>	f= 0000

مجموع ۲۰۱ بیت

۱۸- مرتبهٔ زمانی الگوریتم یافتن مدارهای همیلتونی در یک گراف در بدترین شرایط برابر است با.....

(n!)-1 $O(n^n)$-2 $O(2^n)$-3 $O(n^2 2^n)$-4

جواب : تعداد گره ها در درخت فضای حالت برای الگوریتم یافتن مدارهای همیلتونی برابر است با :

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

که بسیار بدتر از بنایی است با توجه به رابطهٔ بالا الگوریتم از مرتبه $o(n^n)$ میباشد.

۱۹- پیچیدگیرزمانی الگوریتم فلوید در بدترین حالت کدام است؟

۱- $\Theta(n^2)$ ۲- $\Theta(n^3)$ ۳- $\Theta(2^n)$ ۴- $\Theta(n\log n)$

جواب :

```
Void Floyd (int n , float  w[ ] [n ] ,float D [ ] [ n-1 ]){
int  I, j, k
D= w
For (k=0 ؛ k<n  ؛ k ++)
For ( i=0 ؛ I <n ؛ I ++)
For ( j=0 ؛ j < n ؛ j++)
D [i] [i] = min ( D[i] [j]  D[i] [k] + D [k] [J]  ) }
```

می بینم که در الگوریتم بالا ۳ حلقه تو در تو وجود دارد و عمل اصلی محاسبهٔ مقدار می باشد =>

$T(n)=n\times n\times n=n^3 \in \Theta(n^3)$ گزینه ۴

۲۰-پیچیدگی محاسباتی در هر حالت برای الگوریتم حداقل ضربها .... می باشد.

۱- $\Theta(n^2 2^n)$ ۲- $\Theta(n\log n)$ ۳- $\Theta(n^2)$ ۴- $\Theta(n^3)$

جواب : برای حل مسئلهٔ فروشنده دوره گرد با برنامه نویسی پویا فرض های زیر را در نظر می گیریم.

V=[مجموعهٔ همه رئوس]

A=v زیر مجموعه ای از

کوتاه ترین مسیر از v_{10} به v_1 که از هر راس در A دقیقًا یک بار عبور کند $D[v_i]$ A که در آن $D[V_I][A]$ به صورت زیر محاسبه می شود.

$D[V_I][A]=minimum(w[I] + D [v_i][A - [v_j])$

$V_j \in A$

و اگر $A = 0$ باشد آن گاه داریم $D[v_I][0]= w[I][1]$ بنا به دستور بالا

از گزینهٔ ۲ محاسبه می شود$D[v_2][\{ v_3 , v_4 \}]$

$D[v_2][\{v_3 ,v_4 \}]= minimum \quad |w[2][3]+ D[v_3][v_4]\}$

$w [2] [4]+ D[v_4][v3\}$

۲۱- فرض کنید $T(n)$ تعداد روش های مختلف پرانتز گذاری حاصل ضرب n ماتریس باشد.آنگاه $T(n)$ کدام است؟

$$1- \quad T(n) = \sum_{i=1}^{n-2} T(i)T(n-i)$$

$$2- \quad T(n) = \sum_{i=1}^{n-1} T(i)T(n-i)$$

$$3- \quad T(n) =\approx \sum_{i=1}^{0} T(i)T(n-i)$$

$$4- \quad T(n) = \sum_{i=1}^{n-2} T(i)T(n-i-1)$$

جواب :با توجه به اینکه ضرب ماتریس ها خاصیت شرکت پذیری دارد بنا بر این مثلاً برای سه ماتریس m1، m2 وm3 می توانیم بنویسیم.

$M=(M1\times M2)\times M3= M1\times(M2\times M3)$

وقتی تعداد ضرب ها با ماتریس ها زیاد شود تعداد حالت های ممکن به شدت افزایش می یابد. بطور کلّی از M را به صورت زیر بنویسیم تعداد حالات را می شود محاسبه کرد.

M=(M1×M2×….Mn)

اگر T(n) تعداد حالت های ممکن برای ضرب ماتریس ها باشد خواهیم داشت:

T(n)=$\sum_{i=1}^{n=1} T(n-i)$ گزینه ۲

که بر اساس n به دست می آیند.به این اعداد، اعداد کاتالان گفته می شود.

۲۲- تعداد فرا خوانی ها برای محاسبه p(3,3) در تابع world series؟ زیر کدام است

```
Float worldseries (int n ,float p,flot q)
{
int m,k,
float p [][n+1];
for(m=1;m<=n;m++)
{
p[0][m]=1;
p[m][0]=0
for(k=1;m <=m-1 ;k++)
p[k][m-k]=p*p[k-1][m-k]+q*p[k][m-k-1];
}
For(m=1;m<=n;m++)
For (k=0;k<n-m;k++)
p[m+k][n-k]=p*p[m][m+k-1]+q *p[ k+m][n-k-1];
return p [n][n];
```

۱-۲۰ ۲-۱۸ ۳-۴۰ ۴-۳۸

جواب : روش کد گزاری هافمن برای صرفه جویی در حافظه مصرفی می باشد و در یک درخت کاراکتر ها چیده می شوند و کد های باینری به آنها اختصاص داده می شود به طوری که حروف با تعداد تکرار بیشتر طول کد کمتری داشته باشند.=> گزینهٔ ۲ جواب است.

۲۴- تعداد درخت های جستجوی دودویی که با ۳ کلید متمایز می توان ساخت کدام است؟

۱-۱۵ ۲-۸ ۳-۵ ۴-۳

جواب : برای مسئلهٔ کوله پشتی صفر و یک و فروشنده دوره گرد و الگوریتم های با مرتبهٔ زمانی نمایی ساخته شده که از جمله الگوریتم های انشعاب و تجدید و عقبگرد هستند، که برای بسیاری از نمونه ها بازدهی دارند. پس الگوریتم های با مرتبهٔ زمانی چند جمله ای هنوز ساخته نشده برای این مسائل امّا احتمال وجود آنها هم هنوز رد نشده. پس گزینهٔ ۳ در مورد آنها صدق میکند.

۲۵- الگوریتم عقبگرد برای مسألهٔ مدارهای همیلتونی دارای پیچیدگی زمانی.... می باشد؟

۱-$o(n!)$ ۲-$o(2^n)$ ۳-$o(n^2 \times \log n)$ ۴-$o(n^n)$

جواب : در الگوریتم مذکور تعداد گره ها در درخت فضای حالت عبارت است از :

$$1 + (n-1) + (n-1)2 + \cdots \ldots + n - 1^{n-1} = \frac{(n-1)^n - 1}{n-2}$$

که بسیار بدتر از نمایی است . با توجه به رابطهٔ بدست آمده برای فضای حالت مرتبه زمانی الگوریتم($o(n^n$ در بدترین شرایط می باشد. گزینهٔ ۴

تشریحی

۱- رابطه بازگشتی زیر را به روش حدس و استقرا حل کنید .

$$T(n) = T\left(\left\lceil \frac{n}{2} \right\rceil\right) + T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + 1$$

حل :

حدس میزنیم تابع رشد از مرتبه O(n) باشد. حال باید اثبات کنیم که ضریب ثابت مثبتی مانند ۳ هست که $T(n) \leq C_n$ باشد.

براساس استقرا ، پایه استقرا :

$$T(n) \leq C\left\lceil\frac{n}{2}\right\rceil + C\left\lfloor\frac{n}{2}\right\rfloor + 1 = C_n + 1 \geq C_n$$

بنابراین حدس اشتباه است و آن را اصلاح میکنیم :

$$T(n) \leq C_n + B$$

حال با استقرا ثابت میکنیم که حدس فوق درست میباشد.

فرض استقرا : به ازای هر k<n :

$$(1) \qquad T(k) \leq C_k + B$$

حکم استقرا : ثابت میکنیم به ازای هر n:

$$(2) \qquad T(n) \leq C_n + B$$

با استفاده از (۱) به ازای $k = \frac{n}{2}$ خواهیم داشت :

$$T(n) \leq C_n + 2B + 1 \leq C_n + B$$

پس به ازای $B \leq -1$ رابطه $T(n) \leq C_n + B$ به ازای هر n برقرار است.

با جایگزینی مقدار ثابت برای B مثلا B=-1 خواهیم داشت :

$$T(n) \leq C_n - 1 \implies T(n) \in O(n)$$

۲- پیچیدگی زمانی الگوریتم مرتب سازی سریع (Quick Sort) را در هر دو حالت بدترین حالت و حالت متوسط تحلیل نمائید.

حل :

در الگوریتم سریع در بدترین حالت T(n) در حالت کلی به صورت زیر میباشد :

$$T(n) = \begin{cases} 0 & n < 1 \\ T(n-1) + n - 1 & n \geq 1 \end{cases}$$

در نتیجه مجموع جملات بازگشتی به صورت $T(n) = \frac{n(n-1)}{2}$ خواهد بود بنابراین از مرتبه زمانی $T(n) \in \theta(n^2)$ میباشد.

در حالت متوسط رابطه بازگشتی زیر برقرار است :

$$a_n = \begin{cases} 0 & n < 1 \\ a_{n-1} + \dfrac{2(n-1)}{n(n+1)} & n \geq 1 \end{cases}$$

در نتیجه برای n های بزرگ :

$$a_{n-1} + \frac{2(n-1)}{n(n+1)} = a_{n-1} + 2\left(\frac{2n-1}{n} - \frac{2n}{n+1}\right)$$

که پس از حل با روش تکرار و جایگزینی در نهایت خواهیم داشت :

$$a_n \leq 2 \log nn$$

در الگوریتم سریع در بدترین حالت T(n) در حالت کلی به صورت زیر میباشد :

$$T(n) = \begin{cases} 0 & n < 1 \\ T(n-1) + n - 1 & n \geq 1 \end{cases}$$

در نتیجه مجموع جملات بازگشتی به صورت $T(n) = \frac{n(n-1)}{2}$ خواهد بود بنابراین از مرتبه زمانی $T(n) \in \theta(n^2)$ میباشد.

در حالت متوسط رابطه بازگشتی زیر برقرار است :

$$a_n = \begin{cases} 0 & n < 1 \\ a_{n-1} + \dfrac{2(n-1)}{n(n+1)} & n \geq 1 \end{cases}$$

در نتیجه برای n های بزرگ :

$$a_{n-1} + \frac{2(n-1)}{n(n+1)} = a_{n-1} + 2\left(\frac{2n-1}{n} - \frac{2n}{n+1}\right)$$

که پساز حل با روش تکرار و جایگزینی در نهایت خواهیم داشت :

$$a_n \leq 2 \log nn$$

در نتیجه :

$$\frac{T(n)}{n+1} \leq 2\log nn \Longrightarrow T(n) \leq 2(n+1)\log nn$$

بنابراین پیچیدگی زمانی برابر است با :

$$T(n) \in \theta(n \log n)$$

در نتیجه :

$$\frac{T(n)}{n+1} \leq 2\log nn \Longrightarrow T(n) \leq 2(n+1)\log nn$$

بنابراین پیچیدگی زمانی برابر است با :

$$T(n) \in \theta(n \log n)$$

۳- نحوه محاسبه $\binom{8}{4}$ را با استفاده از برنامه نویسی پویا نشان دهید.

حل :

n \ K	۰	۱	۲	۳	۴
۰	۱				
۱	۱	۱			
۲	۱	۲	۱		
۳	۱	۳	۳	۱	
۴	۱	۴	۶	۴	۱
۵	۱	۵	۱۰	۱۰	۵
۶	۱	۶	۱۵	۲۰	۱۵
۷	۱	۷	۲۱	۳۵	۳۵
۸	۱	۸	۲۸	۵۶	۷۰

سطر اول

B[0][0]=1

سطر دوم

B[1][0]=1

B[1][1]=1

B[2][0]=1

سطر سوم

B[2][1]= B[1][0] + B[1][1] = 1 + 1 = 2

B[2][2]=1

۴- مسئله یافتن حداقل تعداد ضرب اسکالر لازم در ضرب زنجیری ماتریسها را در نظر بگیرید:

الف. مسئله را به روش برنامه نویسی پویا بنویسید.(تابع هدف و اصل بهینگی را تعریف کنید)

ب.الگوریتمی کامل به روش برنامه نویسی پویا بنویسید.

ج.الگوریتم را بر روی نمونه ورودی زیر بکار ببرید و حداقل تعداد ضرب های لازم را به دست آورید.

$$A_{20\times2}B_{2\times30}C_{30\times12}D_{12\times8}$$

جواب : از یک ماتریس M[n][n] استفاده می کنیم که n تعداد ماتریس هایی استکه می خواهیم در یکدیگر ضرب شوند. خانه های این ماتریس با مقادیر زیر پر می شوند.

(برای j) I حداقل تعداد ضرب های لازم برای ضرب

$M[I][J]=A_J$ تا A_I

M[I] [j]= 0 (i=j)

ضرب ۴ ماتریس A B C D به صورت بازگشتی یکی از سه حالت زیر است یعنی اولین نقطۀ جدا کننده بعد از A بابعد از B یا بعد از C باشد.

A(BCD)

(AB)(CD)

(ABC)D

پس از به دست آوردن تعداد حداقل ضرب ها برای هر یک از سه حالت مذکور می توان فرمول اصلی حل مسئله را به صورت زیر نوشت:

(اگر I=J):

$M[i][j]=0$

$M[i][j]=\min(m[i][k] + M[k+1][j] + d_{i-1} d_k d_j)$

(اگر i<j) $i \le k \le j-1$

پرانتز گزاری برای ضرب بهینه به صورت زیر است

A((BC)D)

که تعداد ضرب ها برابر است با

۲× ۳۰×۱۲=۷۲۰

۲×۱۲×۸ =۱۹۲ مجموعاً۱۲۳۲

۲۰×۲×۸=۳۲۰

۵- الگوریتم عقبگرد برای مسئله حاصل جمع زیر مجموعه ها را بنویسید؟

(تعیین همه ترکیبات اعداد صحیح موجود در یکمجموعه n عدد صحیح ، به طوری که حاصل جمع آنها مساوی مقدار معین w شود.)

جواب :

```
Void sum of subsets ( index I , int weight , int total )
{
If ( promising (i))
If ( weight == w )
```

```
Cout << include [1] through include [i];
Else
{
Include [i+1] = "yes";
Sum of subsets (i+1 ,weight + w[i+1] , total – w[i+1]);
Include [i+1] = "No" ;
Sum of subsets (i+1 ,weight , total– w[i+1]);
}
}
Bool promising ( index I )
{
Return((weight +total>=W)&&(weight==W ||weight +W[i+1]<=W));
}
```