

2. مرتبه زمانی قطعه کد زیر کدام چیست؟

```
for (i=; i<=n; i+=2)
    for (j=1; j>i; j++)
        x++;
```

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

جواب: گزینه (ب)

برای به دست آوردن مرتبه زمانی بدترین حالت را در نظر می گیریم، که در بدترین حالت شرط حلقه دوم به صورت $j > n$ خواهد بود. لذا حلقه اول n بار حلقه دوم را تکرار خواهد کرد. از طرفی حلقه دوم در بدترین حالت n بار دستور خود را اجرا می کند لذا $n * n = n^2$ بار آخرین دستور اجرا خواهد بود پس مرتبه زمانی برابر است با $O(n^2)$

4. تابع پیچیدگی زمانی تابع زیر کدام است ؟

```
Void f(int a[], int n){
    if (n==1) return a[0];
    f(a,n-2);
    a[n-1]=a[n-2];
    f(a,n-2);
}
```

$$\begin{array}{ll} T(n) = 1 & n=1 \\ 2T(n-2)+T(n-1) & n>1 \end{array} \quad ۱.$$

جواب گزینه: (ب)

همانطور در کد نشان داده شده در تابع f دو بار تابع با پارامتر $n-2$ فراخوانی می شود و ما بین آن یک دستور اجرا خواهد شد پس می توان نوشت

$$T(n) = 2T(n-2) + 1$$

$$\begin{array}{ll} T(n) = 1 & n=1 \\ 2T(n-2)+1 & n>1 \end{array} \quad ۲.$$

$$\begin{array}{ll} T(n) = 1 & n=1 \\ T(n-2) + T(n-1) + 1 & n>1 \end{array} \quad ۳.$$

$$\begin{array}{ll} T(n) = 1 & n=1 \\ T(n-2) + 1 & n>1 \end{array} \quad ۴.$$

6. کدام گزینه صحیح است؟

$$T(n) = 3T\left(\frac{n}{5}\right) + n^2 \in \theta(n^{\log 53}). \quad ۲.$$

$$T(n) = T\left(\frac{n}{2}\right) + n \in \theta(n \log n). \quad ۴.$$

$$T(n) = 2T(n-1) + 1 \in \theta(2^n/2). \quad 1.$$

$$T(n) = T(n-1) + 1 \in \theta(n). \quad ۳.$$

جواب گزینه (سه)

$$T(n) = aT\left(\frac{n}{b}\right) + F(n) \quad (F(n) \in n^k)$$

در صورتی که n یک عدد طبیعی و $b > 1$ و $a \geq 1$ باشد، به صورت زیر بیان می شود:

$$T(n) = \begin{cases} \theta(n^{\log_b a}) & a > b^k \\ \theta(n^k \log n) & a = b^k \\ \theta(n^k) & a < b^k \end{cases}$$

در گزینه ۲) $a=3$ ، $b=5$ و $f(n)=n^2$ ، $k=2$ ، لذا $a < b^k$ پس $\theta(n^2)$

در گزینه ۴) $a=1$ ، $b=2$ و $f(n)=n$ ، $k=1$ ، لذا $a < b^k$ پس $\theta(n)$

$$T(n) = aT(n-b) \rightarrow \theta(a^{n/b})$$

if $(a=1) \rightarrow \theta(n)$

در گزینه ۱) $a=2$ ، $b=1$ لذا $\theta(2^n)$

در گزینه ۳) $a=1$ لذا $\theta(n)$

8. آرایه نه عنصری a مفروض است. اگر این آرایه به روش مرتب سازی سریع مرتب شود، خروجی تابع partition در مرحله اول چیست؟

۱۴	۲۵	۷	۱۸	۵	۳۲	۴۱	۲	۹
----	----	---	----	---	----	----	---	---

۱.

۵	۹	۷	۲	۱۴	۳۲	۴۱	۱۸	۲۵
---	---	---	---	----	----	----	----	----

۲.

۲	۵	۷	۹	۱۴	۱۸	۲۵	۳۲	۴۱
---	---	---	---	----	----	----	----	----

۳.

2	5	7	9	14	18	25	32	41
---	---	---	---	----	----	----	----	----

۴.

۹	۷	۵	۲	۱۴	۳۲	۴۱	۱۸	۲۵
---	---	---	---	----	----	----	----	----

i در عنصر بزرگ تر از ۱۴ و j در عناصر کوچک تر از ۱۴ متوقف شده و عناصر تعویض می شوند.

۱۴	۲۵	۷	۱۸	۵	۳۲	۴۱	۲	۹
----	----	---	----	---	----	----	---	---

→

۱۴	۷	۲۵	۱۸	۵	۳۲	۴۱	۲	۹
----	---	----	----	---	----	----	---	---

14	7	5	18	25	32	41	2	9
----	---	---	----	----	----	----	---	---

→

۱۴	۷	۵	۲	۲۵	۳۲	۴۱	۱۸	۹
----	---	---	---	----	----	----	----	---

۱۴	۷	۵	۲	۹	۳۲	۴۱	۱۸	۲۵
----	---	---	---	---	----	----	----	----

→

۹	۷	۵	۲	۱۴	۳۲	۴۱	۱۸	۲۵
---	---	---	---	----	----	----	----	----

10. پیچیدگی زمانی الگوریتم mergesort چقدر است؟

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

جواب گزینه (ب). پیچیدگی زمانی الگوریتم mergesort (ادغامی) برابر با $\theta(n \log n)$ است.

12. اگر برای یافتن بیشترین و کمترین مقدار یک آرایه ۱۰ عنصری از الگوریتم زیر استفاده شود، تعداد مقایسه ها چقدر است؟

```
Void MaxMin(int a[], int low, int high, int & min, int& max){  
if(low ==high) min=max=a[low];  
else if (low==high -1){  
    if(a[low]<a[high]){  
        max = a[high]; min=a[low];}  
    else{  
        max=a[low]; min=a[high];}  
    else{  
        int mid=(low+high)/2, max1, min1;  
        MaxMin(a,low, mid, min, max);  
        MaxMin(a,mid+1,high,min1,max1);  
        If(min1<min) min = min1;  
        If(max1>max) max = max1;  
    }  
}
```

5.۱ 10.۲ 12.۳ 13.۴

جواب گزینه (د).

الگوریتم مربوط به پیدا کردن ماکزیمم و مینیمم در لیست، مربوط به minmax می باشد که رابطه آن به صورت $T(n) = 3n/2 - 2$ می باشد، بنابراین برای ۱۰ عنصر $T(n) = 30/2 - 2 = 13$ مقایسه خواهیم داشت.

نکته: اگر عناصر فرد باشد تعداد مقایسه ها برابر است با $T(n) = 3n/2 - 3/2$

۱۴. در صورتی که متن زیر به روش هافمن کد گذاری شود، کد حرف b کدام خواهد بود؟

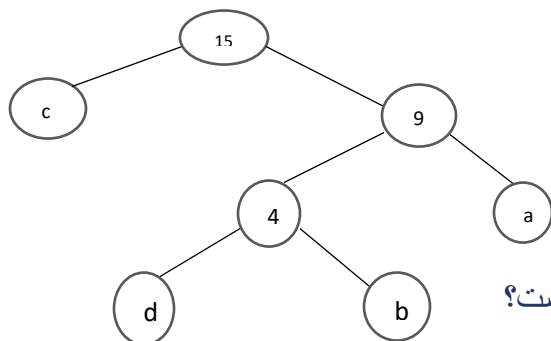
abaacccdbabccccc

100.۱ 001.۲ 01.۳ 101.۴

جواب گزینه (د).

در کد هافمن، دو کاراکتر با تعداد کمتر باهم تشکیل گره ای را داده و مجموع تعداد آنها در آن گره ثبت می شود سپس عدد این گره در لیست تعداد کاراکترها قرر می گیرد. این روند را تا ریشه درخت ادامه می دهیم بعد از تشکیل درخت، از

ریشه یال های سمت چپ ۰ و یال های سمت راست را ۱ می گذاریم پس از آن برای هر کاراکتر از ریشه شروع کرده عدد به دست آمده از ریشه تا آن کاراکتر را از روی یال ها می نویسیم: $a=5$ ، $b=3$ ، $c=6$ ، $d=1$



$a=11$ ، $b=101$ ، $c=0$ ، $d=100$

۱۶. تعداد حالات مختلف ضرب زنجیره ای ۵ ماتریس کدام است؟

۱۰ . ۴ ۳۰ . ۳ ۵ . ۲ ۱۴ . ۱

جواب گزینه (الف)

تعداد حالات ممکن برای ضرب زنجیره ای ماتریس ها از رابطه زیر به دست می آید:

$$T(n) = \sum_{i=1}^{n-1} T(i)T(n-i) = \frac{1}{n} \binom{2(n-1)}{n-1} = \frac{1}{5} \binom{2(5-1)}{5-1} = \frac{1}{5} \frac{8!}{4!4!} = 14$$

۱۸. مرتبه زمانی مساله کوله پشتی صفرو یک با استفاده از روش برنامه نویسی پویا چقدر است؟

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

جواب گزینه (ج)

مرتبه زمانی کوله پشتی صفرو یک در بدترین حالت $\theta(2^n)$ می باشد.

۲۰. دو رشته $X=ABCBDBAB$ و $Y=BDCABA$ را در نظر بگیرید. اگر برای یافتن طولانی ترین زیر رشته مشترک بین X و Y از روش برنامه نویسی پویا استفاده شود، $b[3][3]$ چقدر است؟

۰ . ۱ ۱ . ۲ ۲ . ۳ ۳ . ۴

جواب گزینه (ج)

حل: ابتدا باید تک تک خانه های جدول را پر کنیم.

در گام اول خانه هایی که دارای $i=0$ یا $j=0$ می باشند، با عدد صفر پر شوند.

در گام بعدی اگر در خانه ای $X_i=Y_j$ بود، مثل خانه $C[4,5]$

$$C[4,5] = C[3,4] + 1 = 2 + 1 = 3$$

و اگر در خانه ای $X_i \neq Y_j$ بود، مثل خانه های $C[6,3]$ و $C[2,6]$

$$C[6,3] = \max(C[5,3], C[6,2]) = \max(2, 2) = 2 \uparrow \leftarrow$$

$$C[2,6] = \max(C(1,6), C(2,5)) = \max(1,2) = 2 \leftarrow$$

۲۲. گرافی با ماتریس مجاورت زیر مفروض است. برای رنگ آمیزی این گراف با سه رنگ چند پاسخ وجود دارد؟

۰.۴ ۱.۳ ۲.۲ ۳.۱

```

0  0  0  0
0  0  0  0
0  0  0  0
0  0  0  0

```

جواب گزینه (د)

چون همه راس ها بر هم منطبق هستند و وزن یال بین راس ها ۰ است پس نمی توان گراف را طوری رنگ آمیزی کرد که هیچ دو راس مجاور هم رنگ نباشد.

۲۴. در مساله کوله پشتی صفر و یک. مقدار Bound ، profit ، weight در یک گره مفروض به ترتیب معادل کدام گزینه است؟

۱. حد بالایی از بهره قابل دستیابی، حاصل جمع ارزش قطعات و حاصل جمع اوزان قطعات
۲. حاصل جمع ارزش قطعات، حد بالایی از بهره قابل دستیابی و حاصل جمع اوزان قطعات
۳. حد بالایی از بهره قابل دستیابی، حاصل جمع اوزان قطعات و حاصل جمع ارزش قطعات
۴. حاصل جمع اوزان قطعات، حد بالایی از بهره قابل دستیابی و حاصل جمع ارزش قطعات

جواب گزینه (الف)

Profit: حاصل جمع ارزش قطعات

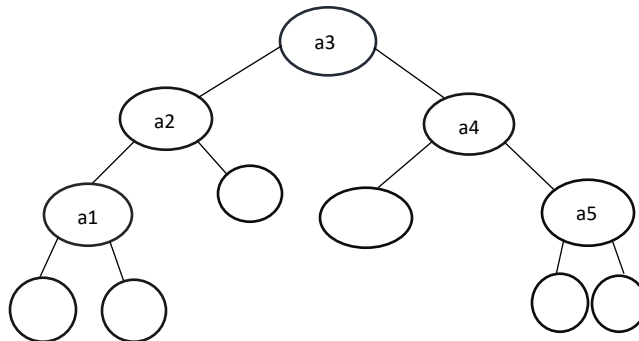
Bound: حد بالا از بهره قابل دستیابی

Weight: حاصل جمع اوزان قطعات

۲. در الگوریتم جستجوی دودویی، متوسط تعداد مقایسه ها در جستجوی موفق و ناموفق برای یک آرایه ۵ عنصری را به کمک درخت تصمیم گیری به دست آورید.

در صورتی که الگوریتم جستجوی دودویی را برای جستجوی عناصر آرایه $A[] = \{a_1, a_2, a_3, a_4, a_5\}$ به کار می ببریم، میانگین تعداد مقایسه ها در جستجوی موفق و ناموفق را به دست می آوریم:

برای محاسبه میانگین جستجوی موفق تعداد گره های هر سطح ، را در شماره سطح آن ضرب می کنیم و جمع می نماییم سپس بر تعداد گره های پر تقسیم می کنیم:



$$\text{میانگین مقایسه های جستجوی موفق} = \frac{1+2 \times 2 + 2 \times 3}{5} = \frac{11}{5} = 2/2$$

برای محاسبه میانگین جستجوی ناموفق تعداد گره های خالی هر سطح، را در شماره سطح آن ضرب می کنیم و جمع می نماییم سپس بر تعداد گره های پر تقسیم می کنیم:

$$\text{میانگین تعداد مقایسه های جستجوی ناموفق} = \frac{2 \times 3 + 4 \times 4}{5+1} = \frac{22}{6} = 3.67$$

* از فرمول های زیر استفاده کردیم:

$$\text{میانگین تعداد مقایسه ها برای جستجوی موفق} = \frac{\sum d(x) \rightarrow \text{مجموع گره های داخلی}}{n \rightarrow \text{تعداد کل گره}}$$

$$\text{میانگین تعداد مقایسه ها برای جستجوی ناموفق} = \frac{\sum dx \rightarrow \text{طول مجموع گره های داخلی}}{n+1}$$

۴. در مساله حاصل جمع زیر مجموعه ها، اگر $n=5$ و $w=12$ باشد، برای w_i های داده شده زیر، با استفاده از تکنیک عقبگرد چند جواب وجود دارد؟ درخت فضای حالت آن را رسم کنید.

$$W_1=2, W_2=5, W_3=7, W_4=10, W_5=12$$

پاسخ:

در مسئله حاصل جمع زیر مجموعه ها، N عدد مثبت و صحیح w_i (وزن ها) و یک عدد صحیح مثبت W داده شده و هدف، یافتن همه ی زیر مجموعه هایی از این اعداد صحیح است که حاصل جمع آن ها برابر W بشود. اگر $weight$ مجموع وزن های جمع آوری شده تاکنون و w_{i+1} وزن قطعه بعدی، $total$ مجموع وزن های پیمایش نشده (باقی مانده) باشد داریم:

$$\rightarrow \begin{cases} weight + w_{i+1} \leq w \\ weight + total \geq w \end{cases} \text{ شروط امید بخش بودن مسئله حاصل جمع زیر مجموعه ها}$$

$$\rightarrow \begin{cases} weight + w_{i+1} > w \\ weight + total < w \end{cases} \text{ شروط ناامید بخش بودن مسئله حاصل جمع زیر مجموعه ها}$$

برای رسم درخت فضای حالت : اگر از ریشه به طرف چپ بریم، یعنی w_1 را انتخاب کرده ایم و اگر به طرف راست بریم، یعنی w_1 انتخاب نشده است. به همین ترتیب اگر از یک گره در سطح ۱، به سمت چپ رفتیم، یعنی w_2 را انتخاب کرده ایم و اگر به طرف راست رفتیم، آن را انتخاب نکرده ایم و... هر مسیر از ریشه به برگ معرف یک زیر مجموعه است.

۱. تابع زیر را در نظر بگیرید. پیچیدگی زمانی تابع زیر کدام است؟

```
i=n
while(i>1)
{
J=1
While(J=n)
J=2*J;
i--;
}
```

۱. $O(\log \log n)$ ۲. $O(\log n)$ ۳. $O(n)$ ۴. $O(n \log n)$

پاسخ گزینه (د)

در حلقه while اگر شمارنده حلقه در داخل حلقه ضرب یا تقسیم بر عدد ثابتی مانند a شود مرتبه زمانی آن حلقه $\theta(\log_a n)$ می شود. با توجه به اینکه شمارنده حلقه while دوم، در داخل حلقه بر ۲ ضرب می شود مرتبه زمانی الگوریتم برابر $\theta(\log_2 n)$ است. از طرفی شمارنده حلقه while اول در داخل حلقه، در هر بار اجرا یک واحد کاسته می شود با توجه به $i=n$ ، پس بار اجرا خواهد شد. با توجه به اینکه اگر حلقه ای در داخل حلقه دیگر اجرا شود پیچیدگی زمانی آن دو حلقه در هم ضرب می شوند، پیچیدگی زمانی تابع برابر است با $O(n \log n)$

۳. در روش جستجوی خطی برای یافتن یک عنصر، درون آرایه n عنصری به چه تعداد مقایسه نیاز دارد؟

۱. n ۲. n+1 ۳. n-1 ۴. 1

پاسخ گزینه (الف)

Ⓒ بهترین حالت \leftarrow عنصر مورد جستجو (x) در ابتدای لیست باشد $T(n) \in O(1)$

Ⓒ بدترین حالت \leftarrow X در انتهای لیست باشد $T(n) \in O(n)$

Ⓒ حالت متوسط $\leftarrow \frac{\text{مجموع مقایسات}}{\text{تعداد عناصر آرایه}} = \text{متوسط مقایسات}$

لذا در بدترین حالت نیاز به n مقایسه داریم.

۵. مرتبه پیچیدگی زمانی تابع زیر چیست؟

```
Int f(int n)
{
  If (n==1)
  Return 1;
Else
Return (n+f(n-1));
}
```

۱. n^2 ۲. n ۳. $n \log n$ ۴. $n^2 \log n$

پاسخ گزینه (ب)

همانطور که در تابع دیده می شود در هر بار اجرای تابع بر اساس $f(n-1)$ ، یک واحد از n کاسته شده و تابع دوباره فراخوانی می شود. پس تابع n بار اجرا خواهد شد و پیچیدگی زمانی آن برابر با n می باشد.

۷. داده ها ۵۴ را با چندمین مقایسه در آرایه مقابل با روش bin search پیدا می کنیم؟
(2,8,10,18,18,20,30,54)

۱. 3 ۲. 4 ۳. 7 ۴. 1

پاسخ گزینه (ب)

تعداد مقایسه ها را برای هر یک از عناصر به دست می آوریم. در جستجوی دودویی عنصر وسط مقایسه می شود اگر عدد مورد جستجو کوچکتر باشد عناصر سمت چپ آن، و اگر بزرگتر باشد عناصر سمت راست آن بررسی می شوند. تعداد مقایسه ها برای رسیدن به هر عنصر در بالای آن نوشته شده است:

3	2	3	1	3	2	3	4
2	8	10	18	18	20	30	54

۹. بهترین الگوریتم ضرب دو عدد n بیتی دارای کدام یک از پیچیدگی ها می باشد؟

۱. $O(n)$ ۲. $O(n^2)$ ۳. $O(n \log n)$ ۴. $O(n^{\log_2 3})$

پاسخ گزینه (د)

با توجه به مسئله ضرب اعداد صحیح بزرگ بهترین الگوریتم برای ضرب دو عدد n بیتی داری پیچیدگی زیر است:

$T(n) \in \theta(n^{\log_2 3})$

۱۱. می خواهیم از روش استراسن، دو ماتریس را در هم ضرب کنیم، کدام گزینه صحیح نیست؟

۱. پیچیدگی مسئله کاهش می یابد.
۲. استراسن از روش تقسیم و حل استفاده می کند.
۳. استراسن یک روش خاص می باشد و همواره کاربرد ندارد.
۴. استراسن مرتبه ضرب را از n^3 کاهش می دهد.

پاسخ گزینه (ج)

استراسن از روش تقسیم و حل استفاده می کند و پیچیدگی آن از n^3 بهتر است و پیچیدگی مسئله ضرب ماتریس ها را کاهش می دهد و کاربرد فراوانی دارد.

۱۳. با توجه به الگوریتم زمان بندی با مهلت معین، اگر مهلت ها و زمان ها و بهره ها به صورت جدول زیر باشد، بیشترین قابل دستیابی ، کدام یک از موارد زیر است؟

کار	مهلت	بهره
۱	۲	۳۰
۲	۱	۳۵
۳	۲	۲۵
۴	۱	۴۰

۱. 60 ۲. 90 ۳. 65 ۴. 70

پاسخ گزینه (د)

حل: ابتدا کارها را بر اساس سود به صورت نزولی مرتب می نماییم سپس با استفاده از جدول الگوریتم زمانبندی با مهلت معین با انتخاب کارها به ترتیب از بالا به پایین، مجموعه امکانپذیر را به دست می آوریم. کارها به ترتیب بررسی می شوند اگر مطابق با مهلت مشخص شده برای آن کار، زمانی پیدا شد که کارهای دیگر اجرا نمی شود آن کار را به مجموعه اضافه میکنیم، مثلا اگر مهلت کاری ۳ باشد و دو کار با مهلت های ۱ و ۳ قبلا انتخاب شده باشند، کار با مهلت ۱ به ناچار در زمان ۱ اجرا می شود کار با مهلت ۳ می تواند در زمان های ۲ یا ۳ اجرا شود پس در یکی از زمان ها اجرا می شود و کار جدید با مهلت ۳ می تواند در زمان بعدی اجرا شود. مثال دیگر: اگر ۲ کار قبل با مهلت های ۱ و ۳ انتخاب شده باشند کار جدید با مهلت ۱ نمی تواند انتخاب شود زیرا قبلا زمان ۱ به کار دیگری داده شده است. منظور از مهلت یعنی

کار می تواند در یکی از زمان ها تا آن مهلت اجرا شود.
 بهره کار بعلاوه بهره کار ۴ برابر با ۷۰ می شود.

امکانپذیر	مجموعه	سود	مهلت	کار
هست	{۴}	۴۰	۱	۴
نیست	{۴}	۳۵	۱	۲
هست	{۴، ۱}	۳۰	۲	۱
نیست	{۴، ۱}	۲۵	۲	۳

۱۵. فرض کنید A و B و C و D به ترتیب ماتریس های ۱۳×۵ ، ۸۹×۳ ، ۳×۸۹ و ۳×۳۴ باشند حداقل تعداد ضرب مورد نیاز برای محاسبه ی ماتریس M=ABCD است؟

۴۰۵۵ .۱ ۲۸۵۶ .۲ ۳۴۲۵ .۳ ۴۰۵۲۱ .۴

پاسخ گزینه (ب)

$$A_{13 \times 5} * B_{5 \times 89} * C_{89 \times 3} * D_{3 \times 34}$$

ابتدا ضرب هایی را انجام می دهیم که بعد مشترک بزرگتری داشته باشد.

$$\left. \begin{aligned} B_{5 \times 89} * C_{89 \times 3} &= 5 * 89 * 3 = 1335 \Rightarrow E_{5 \times 3} \\ &\Rightarrow A_{13 \times 5} * E_{5 \times 3} * D_{3 \times 34} \\ A_{13 \times 5} * E_{5 \times 3} &= 13 * 5 * 3 = 195 \Rightarrow F_{13 \times 3} \\ &\Rightarrow F_{13 \times 3} * D_{3 \times 34} \\ F_{13 \times 3} * D_{3 \times 34} &= 13 * 3 * 34 = 1326 \Rightarrow G_{13 \times 34} \end{aligned} \right\} 1335 + 195 + 1326 = 2856$$

۱۷. کدام یک از الگوریتم های زیر بر اساس روش برنامه نویسی پویا نیست؟

۱. ایجاد درخت جستجوی دودویی بهینه
۲. الگوریتم پریم
۳. ضرب زنجیره ی ماتریس ها
۴. فروشنده دوره گرد

پاسخ گزینه (ب)

روش برنامه نویسی پویا برای حل الگوریتم های زیر کاربرد دارد

- زنجیره ضرب ماتریس ها (حداقل ضرب ها)
- محاسبه ترکیب k مولفه از n مولفه (محاسبه ضریب چند جمله ای)
- الگوریتم فلوید (برای کوتاه ترین مسیر)
- درخت جستجوی دودویی بهینه
- مسئله فروشنده دوره گرد (TSP)
- مسئله کوله پشتی صفر و یک
- مسئله ویرایش رشته ها
- طولانی ترین زیررشته مشترک (LSC) و

۱۹. پیچیدگی الگوریتم فیبوناتچی که به روش برنامه نویسی پویا پیاده سازی شده است متعلق به کدام مرتبه زمانی است؟ آیا این پیچیدگی را باز می توان کاهش داد؟

۱. Logn و خیر ۲. Logn و بله ۳. n و بله ۴. n و خیر

پاسخ گزینه (د)

پیچیدگی الگوریتم های پویا، مسئله بهینه سازی موضوعی کلیدی است. الگوریتم های حریصانه هم اغلب برای مسائل بهینه سازی کاربرد دارند.

21. در مسئله رنگ آمیزی گراف اگر n نشان دهنده تعداد رئوس گراف و m تعداد رنگ ها باشد، درخت فضای حالت حداکثر چند گره خواهد داشت؟

۱. $m^{n+1} - 1/m$ ۲. $m^{n+1} - 1/m - 1$ ۳. $m^{n+1} + 1/m$ ۴. $m^{n+1} + 1/m + 1$

پاسخ گزینه (ب)

تعداد گره ها در درخت فضای حالت برای این الگوریتم برابر است با:

$$1 + m + m^2 + \dots + m^n = \frac{m^{n+1} - 1}{m - 1}$$

23. کدام یک از روش های پیمایش زیر برای گراف ها وجود دارد؟

۱. BFS ۲. DFS ۳. DFS و BFS ۴. BFS و DFS و LFS

پاسخ گزینه (ج)

اصولا ۲ روش جستجوی اصلی برای پیمایش گراف ها در حالت کلی وجود دارد:

۱. DFS ← جستجوی عمقی ← مربوط به روش عقبگرد می باشد.
۲. BFS ← جستجوی ردیفی (سطحی) ← مربوط به روش انشعاب و تحدید می باشد.

۲۵. گزینه نادرست کدام است؟

۱. تمام مسائل P به وسیله یک الگوریتم غیر قطعی در زمان چند جمله ای حل می شوند.
۲. تمام مسائل NP به وسیله یک الگوریتم غیر قطعی در زمان چند جمله ای حل می شوند.
۳. تمام مسائل NP-hard به وسیله یک الگوریتم غیر قطعی در زمان چند جمله ای حل می شوند.
۴. تمام مسائل NP-complete به وسیله یک الگوریتم غیر قطعی در زمان چند جمله ای حل می شوند.

پاسخ گزینه (ج)

NP_Complete: مسائلی که هم NP هستند و هم NP_Hard یعنی هم باید در صورت دادن یک جواب بتوان

درستی جواب رو در مدت زمان چند جمله ای بررسی کرد و هم بتوان بقیه مسائل NP رو به این مسئله کاهش داد.

تشریحی

۱. الگوریتم فلوید را با ذکر یک مثال توضیح داده و از نظر پیچیدگی زمانی بررسی نمایید؟

الگوریتم فلوید (Floyd) برای محاسبه کوتاه ترین مسیر از هر راس در یک گراف موزون به رئوس دیگر به کار می رود.

در این روش ماتریس های D^0 تا D^n را به ترتیب به دست می آوریم.، جواب مسئله ماتریس D^n خواهد بود و در آن کوتاه ترین مسیر ها برای ۲ گره مشخص شده است.
ماتریس D^0 همان ماتریس مجاورت گراف می باشد که در آن:

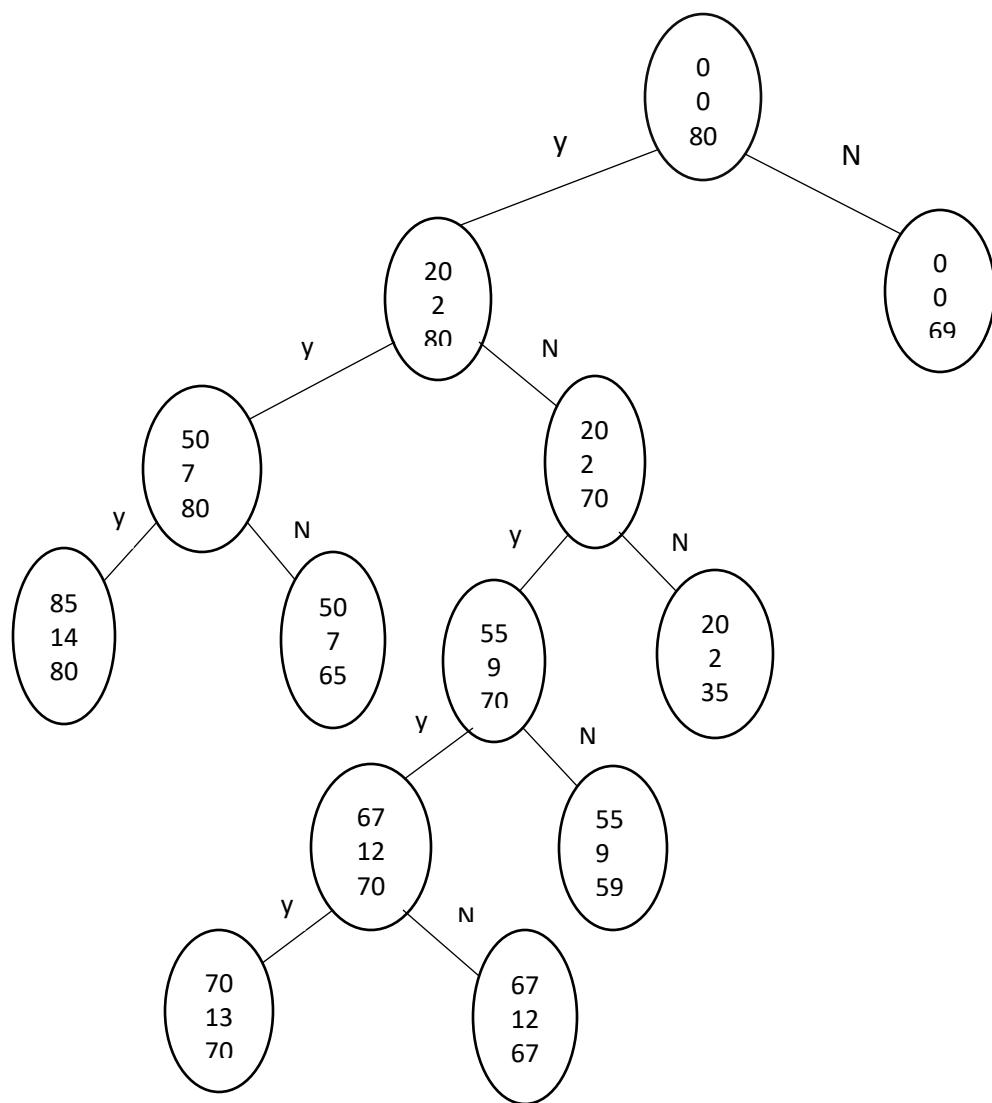
$$D^0 = w(i,j) = \begin{cases} \text{وزن یال} & \text{اگر یالی بین } v_i \text{ و } v_j \text{ باشد} \\ \infty & \text{اگر یالی بین } v_i \text{ و } v_j \text{ نباشد} \\ 0 & \text{اگر } i = j \text{ باشد} \end{cases}$$

فرمول کلی زیر برای الگوریتم فلوید به کار می رود:

$$D^k(i,j) = \begin{cases} 0 & \text{اگر } i = j \\ \min_{i \leq k \leq j} (D^{k-1}(i,k) + D^{k-1}(k,j), D^{k-1}(i,j)) & \text{اگر } i \neq j \end{cases}$$

۳. برای مسئله کوله پشتی صفر و یک با ظرفیت کوله پشتی $W=13$ و پنج قطعه داده شده به صورت جدول زیر با استفاده از روش انشعاب و تحدید مسئله را حل کرده و درخت فضای حالت هرس شده را برای آن رسم نمایید.

i	P_i	W_i	
1	20	2	10
2	30	5	6
3	12	3	4
4	3	1	3
5	35	7	5



۵. کلاس NP و مجموعه مسائل مربوطه را تعریف کنید و دو مثال بزنید.

برای مسائل کلاس NP ، کامپیوتر علاوه بر توانایی اجرای دستورهای معین، باید قادر باشد دستورات نامعین را نیز اجرا کند.

معمولا دستورهای زیر به الگوریتم های معین اضافه می شود تا به الگوریتم نامعین تبدیل شود.

الف) تابعی که یکی از عناصر مجموعه S را به دلخواه انتخاب می کند. (مرحله حدس زدن)

ب) حدس، انتخاب شده و مجموعه S ورودی این تابع می بنشد. خروجی این تابع ، پایان موفق یا ناموفق

عملیات الگوریتم را اعلام می کند. (مرحله تصدیق)

NP مجموعه تمام مسائل تصمیم گیری است که به وسیله ی یک الگوریتم غیر قطعی در زمان چند جمله ای حل

می شوند. مثال : فروشنده ی دوره گرد، حاصل جمع زیر مجموعه ها، رنگ آمیزی گراف ها و کوله پشتی ۰ و ۱

و...

مسائلی که الگوریتم کارآمد (چندجمله ای) برای آن ها ابداع نشده است، ولی غیر ممکن بودن آن نیز هنوز به اثبات

نرسیده را مسائل **NP کامل** می گویند.