بخش اول: آنالیز زمانی

فصل اول

«پیچیدگی زمانی الگوریتم‌ها»

تست‌های تألیفی فصل اول

مثال۱: کدام ترتیب برای توابع زیر به‌ازای مقادیر بزرگ n صحیح است؟

۱) $\lg^{\frac{\lg(n^3)}{\lg\lg n}} n < n^{\lg^* n} < n^{2^{\lg(1+\frac{1}{n})^n}}$

۲) $n^{2^{\lg(1+\frac{1}{n})^n}} < \lg^{\frac{\lg(n^3)}{\lg\lg n}} n < n^{\lg^* n}$

۳) $n^{2^{\lg(1+\frac{1}{n})^n}} < n^{\lg^* n} < \lg^{\frac{\lg(n^3)}{\lg\lg n}} n$

۴) $n^{\lg^* n} < \lg^{\frac{\lg(n^3)}{\lg\lg n}} n < n^{2^{\lg(1+\frac{1}{n})^n}}$

☑ **پاسخ:** گزینه «۲» برای 3 تابع فوق خواهیم داشت: (منظور از e عدد نپر با مقدار 2.71828 است.)

$$n^{2^{\lg(1+\frac{1}{n})^n}} = n^{(1+\frac{1}{n})^n} = n^e$$

$$\lg^{\frac{\lg(n^3)}{\lg\lg n}} n = 2^{\lg\lg^{\frac{\lg(n^3)}{\lg\lg n}} n} = 2^{\frac{\lg(n^3)}{\lg\lg n}\lg\lg n} = 2^{\lg(n^3)} = n^3$$

برای nهای بزرگ، مقدار $\lg^* n$ نیز از مقدار اعداد e و 3 بزرگ‌تر است.

مثال۲: چه تعداد از عبارت‌های زیر درست است؟

$$\sum_{i=1}^{n}(\lg i!) \in O(n^2)$$

$$(1+\frac{1}{n})^{n^2} \in O(n^2 \lg n)$$

$$(\lg n!) \in O(n^n)$$

۱) 0
۲) 1
۳) 2
۴) 3

☑ **پاسخ:** گزینه «۱» هیچ‌یک از عبارت‌ها صحیح نیستند. عبارت $\sum_{i=1}^{n}(\lg i!)$ از مرتبه $\theta(n^2 \lg n)$ است. خواهیم داشت:

$$\sum_{i=1}^{n}(\lg i!) = \sum_{i=1}^{n}(\lg i + \lg(i-1) + \lg(i-2) + \cdots) = \sum_{i=1}^{n}\theta(i\lg i) = \theta(n^2 \lg n)$$

عبارت $(1+\frac{1}{n})^{n^2}$ معادل با عبارت e^n خواهد بود. $((\lg n)!)!$ نیز از مرتبه $O(\theta(n\lg n)!)$ می‌باشد.

مثال۳: چه تعداد از گزاره‌های زیر صحیح هستند؟

ـ اگر $f(n) \in o(g(n))$ باشد، $\lg(g(n)) \in \omega(\lg(f(n)))$ خواهد بود.

ـ اگر $f(n) \in O(g(n))$ و $g(n) \in O(h(n))$ باشد، $2^{f(n)} \in O(2^{g(n)+h(n)})$ خواهد بود.

ـ اگر $f(n) \in \theta(g(n))$ و c یک عدد ثابت باشد، رابطه $g^c(n) \in \theta(f^c(n))$ الزاماً برقرار نخواهد بود.

۱) 0		۲) 1		۳) 2		۴) 3

☑ **پاسخ:** گزینه «۱» هیچ‌یک از عبارت‌ها صحیح نیستند. برای عبارت اول با فرض $f(n) = n^n$ و $g(n) = n^{2n}$ با توجه به اینکه لگاریتم این دو عبارت هم‌مرتبه هستند برقرار نمی‌باشد. در عبارت دوم با فرض $4h(n) = 4g(n) = f(n) = 4n$ رابطه $2^{4n} \in O(2^{2n})$ برقرار نخواهد بود. در رابطه سوم با توجه به $f(n) \in \theta(g(n))$ می‌توان $g(n) \in \theta(f(n))$ و $af(n) \le g(n) \le bf(n)$ را به ازای اعداد ثابت a و b نتیجه گرفت. در نتیجه رابطه $a^c f^c(n) \le g^c(n) \le b^c f^c(n)$ صحیح خواهد بود که برقرار بودن رابطه $g^c(n) \in \theta(f^c(n))$ را نتیجه می‌دهد.

مثال۴: رابطه بازگشتی معادل با تعداد کاراکترهای * چاپ شده کد زیر در کدام گزینه آمده است؟

```
void f(int n) {
   if (n>1) {
      j=0;
      for (i=1 ; i<=n ; i=i+2) {
         j=j+1;
         for (k=1 ; k<=j ; k=k+k) {
            print("*"); }}
      f(j);
}}
```

۱) $f(n)=f(n-2)+\theta(n)$

۲) $f(n)=f(\frac{n}{2})+\theta(n\lg n)$

۳) $f(n)=f(n-2)+\theta(n\lg n)$

۴) $f(n)=f(\frac{n}{2})+\theta(n)$

☑ **پاسخ:** گزینه «۲» حلقه با شمارنده i به تعداد $\frac{n}{2}$ مرتبه تکرار می‌شود و به ازای هر تکرار یک واحد به مقدار j اضافه می‌شود و $\lg j$ مرتبه کاراکتر * چاپ می‌گردد. در خط آخر نیز با توجه به مقدار $j=\frac{n}{2}$ تابع $f(\frac{n}{2})$ فراخوانی خواهد شد. تعداد *های چاپ شده نیز برابر خواهد بود با:

$$\sum_{j=1}^{\frac{n}{2}}\lg j\in\theta(n\lg n)$$

مثال۵: تعداد دفعات چاپ کاراکتر * در کد زیر از چه مرتبه‌ای است؟

```
void F(int n) {
   if (n == 0)
      cout << "*";
   else
      for (int i = 0 ; i < n ; i++)
         F(i);
}
```

۱) $\theta(2^n)$

۲) $\theta(n!)$

۳) $\theta(\frac{4^n}{n\sqrt{n}})$

۴) $\theta(n^2)$

☑ **پاسخ:** گزینه «۱» اگر G_n نشان‌دهنده تعداد کاراکترهای * چاپ شده با فراخوانی تابع $F(n)$ باشد، رابطه بازگشتی معادل با G_n به‌شکل زیر است:
(برای حذف سیگما کافی است اختلاف دو جمله متوالی را محاسبه نماییم.)

$$G_n=\sum_{i=0}^{n-1}G_i\Rightarrow G_n-G_{n-1}=\sum_{i=0}^{n-1}G_i-\sum_{i=0}^{n-2}G_i=G_{n-1}\Rightarrow G_n=2G_{n-1}=2^nG_0=2^n\in\theta(2^n)$$

مثال۶: مقدار نهایی **count** از چه مرتبه‌ای است؟

```
count = 0;
for (i=1 ; i<=n ; i=i+1)
   for (j=1 ; j<=n ; j=j+j)
      for (k=1 ; k<=n ; k=k+i)
         count = count + 1;
```

۱) $\theta(\lg n\lg\lg n)$

۲) $\theta(n\lg^2 n)$

۳) $\theta(n\lg n\lg\lg n)$

۴) $\theta(n^2\lg n)$

☑ **پاسخ:** گزینه «۲» شمارنده i به تعداد n مرتبه تکرار می‌شود. شمارنده j مستقل از i به تعداد $\lg n$ مرتبه تکرار می‌شود. متغیر k به متغیر i وابسته است و به ازای هر مقدار از i به تعداد $\frac{\lg n}{i}$ مرتبه تکرار خواهد شد. مقدار count از رابطه زیر قابل محاسبه است:

$$count=\sum_{i=1}^{n}\sum_{j=1}^{\lg n}\sum_{k=1}^{\frac{n}{i}}1=n\sum_{i=1}^{n}\sum_{j=1}^{\lg n}\frac{1}{i}=n\lg n\sum_{i=1}^{n}\frac{1}{i}=n\lg n\ln n\in\theta(n\lg^2 n)$$

مثال ۷: مقدار sum در پایان کد زیر از چه مرتبه‌ای خواهد بود؟

```
sum = 0;
for (i=n+1;i<=2n;i++)
   for (j=1;j<=i;j++)
      sum = sum + 1/j;
```

۱) $\theta(n\ln n)$

۲) $\theta(\ln n)$

۳) $\theta(n\ln\ln n)$

۴) $\theta(n)$

پاسخ: گزینه «۱» می‌دانیم $\sum_{i=1}^{m}\frac{1}{i}=\ln m$. در نتیجه مقدار sum در هر مرحله از اجرای حلقه با شمارنده j برابر $\ln i$ خواهد شد. با توجه به اینکه مقدار i بین $n+1$ تا $2n$ است مقدار نهایی sum از مرتبه زیر خواهد بود:

$$sum=\sum_{i=n+1}^{2n}\ln i\in\theta(n\ln n)$$

مثال ۸: تابع زمانی کد زیر از چه مرتبه‌ای است؟

```
for (i=1 ; i<n ; i=i+i)
   for (j=0 ; j<i ; j=(i+j+1)/2)
      print("*");
```

۱) $\theta(\lg n\lg\lg n)$

۲) $\theta(n\lg n)$

۳) $\theta(n\lg\lg n)$

۴) $\theta(\lg^2 n)$

پاسخ: گزینه «۴» شمارنده i مقادیر توان‌های عدد 2 که کوچکتر از n هستند را می‌پذیرد. شمارنده j نیز برای هر مقدار از i بـه تعـداد $\lg i$ مرتبـه تکرار می‌شود. در نتیجه تعداد دفعات تکرار دستور print("*") برابر است با:

$$s=\sum_{i=1}^{\lg n}\lg 2^i=\sum_{i=1}^{\lg n}i=\frac{\lg n(\lg n+1)}{2}\in\theta(\lg^2 n)$$

مثال ۹: مقدار نهایی k در کد زیر از چه مرتبه‌ای است؟

```
i = 1;
k = 1;
n = m;
while (i<n) {
   k = k + i;
   i = i + i;
   n = n / 2;
}
```

۱) $\theta(\lg\lg m)$

۲) $\theta(m)$

۳) $\theta(\lg m)$

۴) $\theta(\sqrt{m})$

پاسخ: گزینه «۴» در هر تکرار از حلقه while، مقدار i به k اضافه، مقدار i دو برابر و مقدار n نصف می‌شود. در لحظه خروج از حلقـه مقـدار i و n برابر با $2^{\frac{\lg m}{2}}$ خواهد بود. مقدار k برابر خواهد بود با:

$$k=1+\sum_{i=0}^{\frac{\lg m}{2}}2^i=1+2^{\frac{\lg m}{2}+1}-1=2\sqrt{m}=\theta(\sqrt{n})$$

مثال ۱۰: جواب معادله بازگشتی $2a_{n+2}=3a_n+a_{n+1}$ کدام یک از گزینه‌های زیر است؟

۱) $a_n=\frac{2}{3}+\frac{1}{3}(-1)^n$　۲) $a_n=\frac{3}{5}+\frac{2}{5}(-\frac{3}{2})^n$　۳) $a_n=\frac{1}{5}+\frac{4}{5}(-\frac{2}{3})^n$　۴) $a_n=1+(-1)^n$

پاسخ: گزینه «۲» معادله مذکور یک معادله همگن با ضرایب ثابت می‌باشد بنابراین برای حل آن کافیست معادله مشخصه آن را بنویسیم:

$$2r^2-r-3=0\Rightarrow r_1=1,r_2=-\frac{3}{2}\Rightarrow a_n=c_1(1)^n+c_2(-\frac{3}{2})^n=c_1+c_2(-\frac{3}{2})^n$$

مثال ۱۱: از رابطه بازگشتی زیر مقدار $T^2(n)$ کدام است؟

$$T^2(n) = 5T^2(n-1) - 4T^2(n-2)$$

۱) $c_1 2^n + c_2 4^n$ ۲) $c_1 5^n + c_2 4^n$ ۳) $c_1 5^n + c_2(-4)^n$ ۴) $c_1 + c_2 4^n$

☑ **پاسخ:** گزینه «۴» ابتدا قرار می‌دهیم $A(n) = T^2(n)$، بنابراین معادله داده شده به صورت زیر تبدیل خواهد شد:

$$A(n) - 5A(n-1) + 4A(n-2) = 0$$

معادله مشخصه آن به صورت مقابل است: $r^2 - 5r + 4 = 0 \Rightarrow r_1 = 1, r_2 = 4$

در نتیجه جواب کلی این معادله برابر است با:

$$A(n) = c_1(1)^n + c_2(4)^n \Rightarrow A(n) = c_1 + c_2 4^n \Rightarrow T^2(n) = c_1 + c_2 4^n$$

مثال ۱۲: قطعه برنامه زیر را در نظر بگیرید. کدام گزینه تعداد دفعات اجرای دستور $\mathbf{Sum = Sum + 1}$ را به صورت بازگشتی نشان می‌دهد؟

```
for(i = 1 ; i <= n ; i++)
   for(j = 1 ; j <= i ; j++)
      for(k = 1 ; k <= j ; k++)
            Sum = Sum + 1;
```

(تعداد دفعات اجرای این دستور برای یک n خاص را $h(n)$ در نظر بگیرید)

۱) $h(n) = h(n-1) + n(1) = 1\ ,\ h(1) = 1$ ۲) $h(n) = h(n-1) + n + 1\ ,\ h(1) = 3$

۳) $h(n) = h(n-1) + \frac{n(n+1)}{2}\ ,\ h(1) = 1$ ۴) $h(n) = h(n-1) + \frac{n(n+1)}{2}\ ,\ h(1) = 3$

☑ **پاسخ:** گزینه «۳» اگر $n = 1$ باشد آنگاه هر یک از حلقه‌ها فقط یک‌بار و به ازای $i = j = k = 1$ اجرا می‌شوند بنابراین: $h(1) = 1$

حال اگر $h(n-1)$ را داشته باشیم، به ازای تغییر حد بالای حلقه for i به n حلقه for j به ازای $j = 1, 2, \ldots, n$ اجرا می‌گردد و بنابراین تعداد دفعات اجرای حلقه for k برابر خواهد بود با:

$$1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2}$$

بنابراین داریم:

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

مثال ۱۳: بهترین گزینه برای پیچیدگی زمانی رابطه زیر از چه مرتبه‌ای است؟ (فرض کنید $c \le 2$)

$$\mathbf{T(n) = \lg n\, T(\sqrt{n}) + 1 \quad , \quad T(c) = 1}$$

۱) $O(n^{n \lg n})$ ۲) $O(n^{\lg \lg n})$ ۳) $O(2^{\lg^2 n})$ ۴) $O(2^{\lg^2 \lg n})$

☑ **پاسخ:** گزینه «۴» مقدار جمله $T(n)$ را با تکرار محاسبه می‌نماییم. با توجه به ضریب انشعاب رابطه، پیچیدگی زمانی الگوریتم از مرتبه تعداد جملات آخرین سطر در درخت بازگشت معادل (تعداد جملاتی که برابر 1 می‌شوند) خواهد بود و از مقدار ثابت رابطه صرف‌نظر می‌کنیم. با فرض اینکه n توانی از 2 باشد خواهیم داشت:

$$T(n) = T(2^m) = mT(2^{\frac{m}{2}}) = m(\frac{m}{2}T(2^{\frac{m}{4}})) = \cdots = \frac{m^{\lg m + 1}}{2^{\sum_{i=0} \lg m_i}} T(2^{\frac{m}{m}}) = \frac{m^{\lg m + 1}}{2^{\sum_{i=0} \lg m_i}}$$

$$\Rightarrow T(n) = \frac{\lg^{\lg\lg n + 1} n}{2^{\frac{\lg\lg n(\lg\lg n + 1)}{2}}} = \frac{2^{\lg\lg n(\lg\lg n + 1)}}{2^{\frac{\lg\lg n(\lg\lg n + 1)}{2}}} = 2^{\frac{\lg\lg n(\lg\lg n + 1)}{2}} \in O(2^{\lg^2 \lg n})$$

مرتبه عبارت‌های مربوط به ۳ گزینه دیگر از مرتبه عبارت گزینه (۴) بالاتر است.

مثال ۱۴: کدام گزینه تحلیل بهتری از مرتبه زمانی رابطه بازگشتی $T(n)=\lg nT(\lg n)+\lg^2 n$ است؟

۱) $T(n)\in\theta(n\lg^2 n)$　۲) $T(n)\in\theta(\lg^3 n)$　۳) $T(n)\in\theta(\lg^2 n\lg\lg n)$　۴) $T(n)\in\theta(\lg^2 n)$

پاسخ: گزینه «۴» درنظر بگیرید $n=2^{2^m}$ در این صورت خواهیم داشت:

$$T(2^{2^m})=2^m T(2^m)+2^{2m}=2^m mT(m)+2^m m^2+2^{2m}$$
$$=2^{2m}+2^m m^2+2^m m\lg(m^2)+2^m m\lg(m)\lg\lg(m^2)+\cdots\cong 2^{2m}=\lg^2 n$$

در نتیجه این رابطه از مرتبه زمانی $\theta(\lg^2 n)$ خواهد بود.

مثال ۱۵: با فرض $T(1)=1$ مرتبه رشد تابع مقابل را محاسبه نمایید: $T(n+1)=\sum_{i=1}^{n} iT(i)+n^2$

۱) $\theta(n^3)$　۲) $\theta(n^2)$　۳) $\theta(2^n)$　۴) $\theta(n!)$

پاسخ: گزینه «۴» سعی می‌کنیم سیگما را از رابطه حذف کنیم. خواهیم داشت:

$$T(n+1)-nT(n)=\sum_{i=1}^{n} iT(i)+n^2-nT(n)=\sum_{i=1}^{n-1} iT(i)+(n-1)^2+2n-1$$
$$=T(n)+2n-1\Rightarrow T(n)=nT(n-1)+2n-3=nT(n-1)+\theta(n)\in\theta(n!)$$

مثال ۱۶: پاسخ معادله بازگشتی $F(n)=nF(n-1)+n!$ به ازای چه مقدار برای $F(1)$ برابر است با $F(n)=(n+1)!$؟

۱) 0　۲) 1　۳) 2　۴) -1

پاسخ: گزینه «۳» این تست را می‌توان با عددگذاری حل نمود. به عنوان مثال فرض کنید n برابر 2 باشد، بنابراین داریم:

$$\left.\begin{array}{l} n=2\Rightarrow F(2)=(2+1)!=6 \\ n=2\Rightarrow F(2)=2F(1)+2! \end{array}\right\}\Rightarrow 6=2F(1)+2!\Rightarrow F(1)=2$$

مثال ۱۷: رشد تابع مقابل از چه مرتبه‌ای است؟ (منظور از e عدد نپر است.) $T(n)=(1+\frac{1}{n})T(n-1)+n \quad ; \quad T(1)=1$

۱) $\theta(e^n)$　۲) $\theta(n^3)$　۳) $\theta(n^2)$　۴) $\theta(n^e)$

پاسخ: گزینه «۳» رابطه را به شکل زیر بازنویسی می‌کنیم:

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

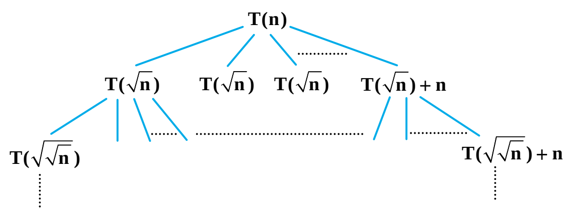
حال از تغییر متغیر $S(n)=\frac{T(n)}{n+1}$ استفاده می‌کنیم. خواهیم داشت:

$$S(n)=S(n-1)+\frac{n}{n+1}\in\theta(n) \Rightarrow T(n)=(n+1)S(n)\in\theta(n^2)$$

مثال ۱۸: با داشتن رابطه مقابل، کدام گزینه درست است؟ $T(n)=\sqrt{n}T(\sqrt{n})+n \ , \ k\le 2: T(k)=1$

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

پاسخ: گزینه «۴» درخت بازگشت این رابطه به صورت زیر خواهد بود:



بنابراین مرتبه زمانی برابر $n\times L$ می‌باشد که L عمق درخت را نشان می‌دهد. برای محاسبه L داریم:

$$n^{(\frac{1}{2})^L}=2 \Rightarrow n^{2^{-L}}=2 \Rightarrow 2^{-L}\times \log_2^n=1$$

بنابراین:

$$\log_2^n=2^L \Rightarrow L=\log_2\log_2^n$$

با یک تغییر متغیر هم می‌توان حل کرد:

$$\frac{T(2)^k}{2^k}=\frac{T(2^{\frac{k}{2}})}{2^{\frac{k}{2}}}+1 \qquad T(2^k)=2^{\frac{k}{2}}+(2^{\frac{k}{2}})+2^k$$

$$\frac{T(2^k)}{2^k}=A(k)\Rightarrow A(k)=A(\frac{k}{2})+1\Rightarrow A(k)+\theta(\log k)\Rightarrow T(2^k)\in\theta(2^k\log k)\Rightarrow T(n)+\theta(n\log\log n)$$

مثال۱۹: مرتبه زمانی رابطه بازگشتی مقابل کدام است؟

$$\mathbf{T(n)=T(\frac{n}{5})+T(\frac{7n}{10})+n}$$

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

پاسخ: گزینه «۱» با استفاده از درخت بازگشت و یا روش جایگذاری خواهیم داشت:

$$T(n)=n+\frac{9}{10}n+\frac{81}{100}n+\ldots$$

بنابراین:

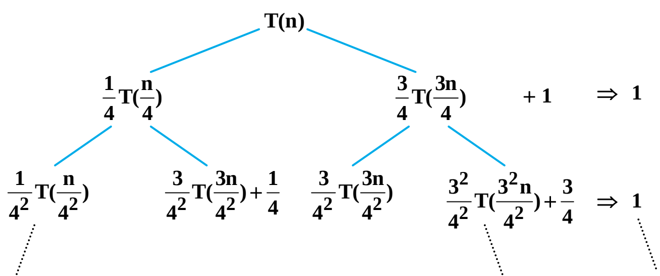
$$T(n)<\sum_{i=0}^{\infty}(\frac{9}{10})^i n \Rightarrow T(n)\in\theta(n)$$

مثال۲۰: مرتبه زمانی رابطه بازگشتی مقابل کدام است؟

$$\mathbf{T(n)=\frac{1}{4}T(\frac{n}{4})+\frac{3}{4}T(\frac{3n}{4})+1}$$

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

پاسخ: گزینه «۲» اگر درخت بازگشت مربوط به این رابطه را در نظر بگیریم، داریم:



بنابراین مرتبه زمانی را عمق درخت تعیین می‌کند، حال اگر عمق درخت را L در نظر بگیریم آنگاه با توجه به شکل درخت داریم:

$$\theta(lon_4^n)<L<\theta(\log_{\frac{4}{3}}^n)\Rightarrow T(n)\in\theta(\log n)$$

مثال۲۱: پیچیدگی زمانی رابطه مقابل از چه مرتبه‌ای است؟

$$\mathbf{T(2n)=T(2[\frac{n}{2}]+1)+T(2[\frac{n}{2}])+\lg n}$$
$$\mathbf{T(2n+1)=T(2n)+n}$$

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

پاسخ: گزینه «۱» کافی‌است از مقدار $T(2n+1)$ را در معادله $T(2n)$ قرار دهیم. خواهیم داشت:

$$T(2n)=T(2[\frac{n}{2}]+1)+T(2[\frac{n}{2}])+\lg n=2T(2[\frac{n}{2}])+\frac{n}{2}+\lg n$$

$$T(2n+1)=T(2n)+n=2T(2[\frac{n}{2}])+\frac{3n}{2}+\lg n$$

سایر مقادیر نیز با استفاده از جملات زوج فراخوانی می‌شوند. در نتیجه رابطه به فرم $T(n)=2T(\frac{n}{2})+\frac{n}{2}+\lg n$ قابل بازنویسی خواهد بود. پیچیدگی زمانی این رابطه از مرتبه $\theta(n\lg n)$ می‌باشد.

مثال۲۲: فرم صریح رابطه بازگشتی مقابل با فرض $T(0)=0$ از چه مرتبه‌ای خواهد بود؟

$$T(n)=4T(\frac{n}{3})+\frac{1}{n}$$

۱) $\theta((\frac{4}{3})^n)$　　۲) $\theta(n\ln n)$　　۳) $\theta((\frac{4}{3})^n \ln n)$　　۴) $\theta(n^{\log_3 4})$

پاسخ: گزینه «۴» رابطه فوق را می‌توان به‌شکل $T(n)=aT(\frac{n}{b})+f(n)$ نوشت. با توجه به اینکه به‌ازای مقادیر a=4 و b=3 قاعده $f(n)\in O(n^{\log_b a-\varepsilon})$ برقرار است، مرتبه زمانی الگوریتم بصورت $T(n)=\theta(n^{\log_b a})=\theta(n^{\log_3 4})$ خواهد بود.

مثال۲۳: مرتبه رابطه بازگشتی مقابل در کدام گزینه آمده است؟

$$T(n)=5T(\frac{n}{3})+n\sqrt{n}$$

۱) $O(n\sqrt{n})$　　۲) $O(n\sqrt{n}\lg n)$　　۳) $O(n^{\log_3 5})$　　۴) $O(n^{\log_3 5}\lg n)$

پاسخ: گزینه «۱» می‌بایست مقدار $n^{\log_3 5}$ را با $n\sqrt{n}$ مقایسه نماییم. درصورتی که $n^{\log_3 5}$ مقدار بزرگتری باشد، رابطه از مرتبه $O(n^{\log_3 5})$ خواهد بود. درصورتی‌که $n\sqrt{n}$ بزرگتری باشد، رابطه از مرتبه $O(n\sqrt{n})$ خواهد بود و درصورت برابر بودن این دو مقدار، رابطه از مرتبه $O(n\sqrt{n}\lg n)$ خواهد بود.

کافی است مقادیر توان این دو عبارت را با هم مقایسه نماییم. توان $n^{\log_3 5}$ برابر $\log_3 5$ و توان $n\sqrt{n}$ برابر $\frac{3}{2}$ می‌باشد. خواهیم داشت:

$$\log_3 5=\frac{1}{2}\log_3 5^2=\frac{1}{2}\log_3 25<\frac{1}{2}\log_3 27=\frac{3}{2}$$

با توجه به بزرگتر بودن مقدار $n\sqrt{n}$، پیچیدگی زمانی الگوریتم از مرتبه $O(n\sqrt{n})$ خواهد بود.

مثال۲۴: رابطه بازگشتی زیر از چه مرتبه‌ای است؟ (فرض کنید $c,d<4$ هستند.)

$$T(n,k)=T(\frac{n}{4},k)+T(\frac{n}{2},\frac{k}{2})+T(n,\frac{k}{4})+(nk)^{\frac{3}{4}}$$

$$T(c,k)=T(4,\frac{kc}{4})\quad,\quad T(c,d)=1\quad,\quad T(n,c)=T(\frac{nc}{4},4)$$

۱) $O((nk)^{\frac{3}{4}})$　　۲) $O((nk)^{\frac{3}{4}}\lg n)$　　۳) $O((nk)^{\lg 3})$　　۴) $O((nk)^{\frac{\lg 3}{2}})$

پاسخ: گزینه «۴» با توجه به شرایط مرزی صورت سؤال، درصورتی‌که رابطه $uv<16$ برقرار باشد، مقدار $T(u,v)$ برابر 1 خواهد شد. مقادیر دو پارامتر در این رابطه تأثیر مجزایی ندارند و تنها حاصلضرب آن‌ها در جواب نهایی تأثیرگذار است. به‌عنوان مثال مقادیر T(i,j) با T(j,i) برابر خواهد بود. زیرا با توجه به شرایط اولیه و رابطه بازگشتی، تأثیر مشابهی دارند. همچنین مقادیری مانند T(mn,k) و T(n,mk) نیز برابر هستند، زیرا مقادیر قسمت ناهمگن معادله بازگشتی آن‌ها معادل است و هر دو عبارت، به 3 جمله دیگر تجزیه می‌شوند که در تمام این جملات، مقدار ناهمگن و حاصلضرب دو پارامتر برابر دارند. در نتیجه می‌توان از تغییر متغیر $S(nk)=T(n,k)$ استفاده نمود. خواهیم داشت:

$$S(nk)=3S(\frac{nk}{4})+(nk)^{\frac{3}{4}}$$

با توجه به اینکه رابطه $\log_4 3>\frac{3}{4}$ و $\log_4 3=\frac{\lg 3}{2}$ برقرار است، برای پیچیدگی محاسباتی رابطه فوق خواهیم داشت:

$$T(n,k)=S(nk)\in O((nk)^{\frac{\lg 3}{2}})$$

اثبات رابطه $\log_4 3>\frac{3}{4}$ به شکل زیر است:

$$\log_4 3=\frac{1}{4}\log_4 3^4=\frac{1}{4}\log_4 81>\frac{1}{4}\log_4 64=\frac{3}{4}$$

مثال۲۵: مقدار نهایی Function(n) از چه مرتبه‌ای خواهد بود؟

```
int Function(int n)
{
  if (n<1) return 1;
  return Function(n/3) + 2*Function(2n/3) +
n*n;
}
```

۱) $O(n^{\log_2 7})$　　۲) $O(n^3)$　　۳) $O(n^2)$　　۴) $O(n^2\log n)$

پاسخ: گزینه «۴» رابطه بازگشتی مربوط به مقدار نهایی تابع به شکل مقابل است:

$$F(n)=F(\frac{n}{3})+2F(\frac{2n}{3})+\theta(n^2)$$

مرتبه رابطه فوق بصورت $F(n)\in O(n^2\log n)$ خواهد بود. با استفاده از درخت بازگشت می‌توان به این رابطه رسید. مجموع مقادیر هر سطر برابر n^2 خواهد بود.

مثال ۲۶: در مسئله برج‌های هانوی، اگر برج‌ها (میله) را در یک ردیف در نظر بگیرید و در نقل و انتقالات تنها مجاز به انتقال از یک میله به میله مجاور باشیم و سایر شرایط نیز مانند شرایط مسئله قبل باشد و a_k تعداد انتقالات لازم برای انتقال k حلقه از میله اول به میله سوم به کمک میله دوم باشد، آنگاه کدام رابطه بازگشتی درباره a_k درست می‌باشد؟

۱) $a_k = 3a_{k-1} + 2$ ۲) $a_k = 2a_{k-1} + 1$ ۳) $a_k = 2a_{k-1} + 2$ ۴) $a_k = 3a_{k-2} + 2$

پاسخ: گزینه «۱» واضح است که $a_1 = 2$، یعنی برای انتقال یک حلقه از میله اول به میله سوم دو حرکت نیاز است. ابتدا آن را به میله دوم منتقل می‌کنیم و سپس آنرا از میله دوم به میله سوم منتقل می‌کنیم.

برای انتقال k حلقه طبق شرایطی که گفته شده، مراحل زیر را طی می‌کنیم.

۱ـ $k-1$ حلقه را از برج اول به برج سوم منتقل می‌کنیم (a_{k-1})

۲ـ یک حلقه باقی‌مانده را از برج اول به برج دوم منتقل می‌کنیم (۱)

۳ـ $k-1$ حلقه را از برج سوم به برج اول منتقل می‌کنیم (a_{k-1})

۴ـ یک حلقه برج دوم را به برج سوم منتقل می‌کنیم. (۱)

۵ـ $k-1$ حلقه را از برج اول به برج سوم منتقل می‌کنیم. (a_{k-1})

بنابراین به طور کلی داریم:
$$a_k = a_{k-1} + 1 + a_{k-1} + 1 + a_{k-1} = 3a_{k-1} + 2$$

مثال ۲۷: در مسئله برج‌های هانوی استاندارد، اگر تعداد حلقه‌ها زوج در نظر گرفته شود و 2n حلقه در برج اول با n اندازه متفاوت دو به دو دسته‌بندی شده وجود داشته باشد و t_{2n} حداقل تعداد انتقالات لازم برای انتقال 2n حلقه از برج اول به برج سوم به کمک برج دوم باشد، آنگاه کدامیک از روابط بازگشتی زیر برای t_{2n} صحیح است؟

۱) $t_{2n} = 2t_{2n-1} + 3$ ۲) $t_{2n} = 2t_{2n-1} + 1$ ۳) $t_{2n} = 2t_{2n-2} + 2$ ۴) $t_{2n} = 2t_{2n-2} + 3$

پاسخ: گزینه «۳» می‌توان ابتدا $2n-2$ حلقه را از برج اول به برج دوم منتقل کرد و سپس دو حلقه آخر که اندازه‌هایشان یکسان است را یکی‌یکی به برج سوم منتقل کنیم، در آخر $2n-2$ حلقه را از برج دوم به برج سوم انتقال می‌دهیم و بدین ترتیب رابطه بازگشتی زیر درباره t_{2n} صادق خواهد بود:
$$t_{2n} = t_{2n-2} + 2 + t_{2n-2} = 2t_{2n-2} + 2$$

مثال ۲۸: کدام گزینه درست است؟

۱) $n^{\frac{1}{10}} \in \Omega(\log n)$ ۲) $\log(n!) \in O(\log n)$

۳) $n^2 + 4n - 10 \in O(n \log n)$ ۴) $1^n + 2^n + 3^n + \cdots + n^n \in \Omega(n^{n+1})$

پاسخ: گزینه «۱» همان‌گونه که قبلاً بیان شد، هر توانی از n مانند n^{ε} (که $\varepsilon > 0$) نسبت به توابع لگاریتمی از رشد بیشتری برخوردار است.

مثال ۲۹: در کدام گزینه توابع از چپ به راست براساس ترتیب رشد نوشته شده‌اند؟

۱) $n \log^n, (1.006)^n, n^2$ ۲) $n(\log n)^2, (1.006)^n, n^2$ ۳) $n^2, n(\log n)^2, (1.006)^n$ ۴) $n(\log n)^2, n^2, (1.006)^n$

پاسخ: گزینه «۴» دقت کنید که رشد توابع نمایی از تمام توابع لگاریتمی و چندجمله‌ای و حاصلضرب آن‌ها بیشتر است.

مثال ۳۰: کدام یک از روابط زیر برقرار است؟

۱) $n^{O(1)} < n^5$ ۲) $\log^5 n > n^{0.4}$ ۳) $3^{\log n} < n^2 \log n$ ۴) $\sqrt{n} < \log^2 n$

پاسخ: گزینه «۳» تابع $3^{\log n}$ از مرتبه $\theta(n^{\log_2^3})$ می‌باشد. در مورد گزینه اول دقت کنید که $O(1)$ می‌تواند شامل هر عدد ثابت بزرگتر از پنج نیز باشد و بنابراین گزینه اول نادرست است.

مثال ۳۱: اگر رابطه بازگشتی T(n) به صورت مقابل تعریف شده باشد آنگاه کدام گزینه جواب رابطه می‌باشد؟ $T(n) = T(n-1) + 2T(n-2)$

۱) $T(n) = c_1 2^n + c_2 n$ ۲) $T(n) = c_1 3^n + c_2 2^n$ ۳) $T(n) = c_1 2^n + c_2(-1)^n$ ۴) $T(n) = c_1 3^n + c_2(-1)^n$

پاسخ: گزینه «۳» در این مثال یک معادله همگن از مرتبه 2 داده شده است که معادله مشخصه آن به صورت زیر می‌باشد:
$$r^2 - r - 2 = 0 \Rightarrow r_1 = 2, r_2 = -1$$

بنابراین جواب کلی آن برابر است با:
$$T(n) = c_1 2^n + c_2(-1)^n$$

آزمون فصل اول

۱ـ کدام عبارت صحیح است؟

۱) $n^2+2n \in O(\frac{n^2}{\log n})$
۲) $n^2+n(n+1)n^2 \in \theta(n^4)$
۳) $n\log n+n^2 \in O(n\mathrm{Log}n)$
۴) $\sum_{i=1}^{n} i^5 \in \theta(n^5)$

۲ـ مرتبه تابع بازگشتی روبرو کدام است؟ $\mathbf{T(n)=2T(\sqrt{n})+Logn}$

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

۳ـ مرتبه زمانی تابع بازگشتی روبرو کدام است؟ $\mathbf{T(n)=T(n-1)+\frac{4}{n}}$

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

۴ـ کدام مورد درست است؟ (فرض کنید ε یک عدد مثبت کوچک باشد)

۱) $2^{\varepsilon}+n^{\varepsilon} \in O(\log n)$
۲) $n^{\varepsilon} \in \theta(\log n)$
۳) $n^2+2n^3 \in O((1+\varepsilon)^n)$
۴) $n! \in O((2+3)^n)$

۵ـ از بین موارد مقابل کدام‌یک صحیح می‌باشد؟ الف) $\mathbf{f(n)+O(f(n)) \in \theta(f(n))}$ ب) $\mathbf{f(n)+\Omega(f(n)) \in \theta(f(n))}$

۱) فقط الف
۲) فقط ب
۳) الف و ب
۴) هیچ‌کدام

۶ـ مرتبه زمانی تابع روبرو کدام است؟ $\mathbf{T(n)=8T(\frac{n}{2})+n^2}$

۱) $\theta(n^2)$
۲) $\theta(n^2\mathrm{Log}n)$
۳) $\theta(n^3)$
۴) $\theta(n^{\mathrm{Log}_8^2})$

۷ـ عمق درخت بازگشت رابطه روبرو کدام است؟ $\mathbf{T(n)=2T(\frac{n}{2})+cn}$

۱) $\theta(\log_2^n)$
۲) $\theta(n)$
۳) $\theta(2n)$
۴) $\theta(n^2)$

۸ـ مرتبه زمانی تابع بازگشتی روبرو کدام است؟ $\mathbf{T(n)=3T(\frac{n}{4})+\theta(n^2)}$

۱) $\theta(n)$
۲) $\theta(n\log n)$
۳) $\theta(n^{\log_4^3})$
۴) $\theta(n^2)$

۹ـ مرتبه زمانی تابع روبرو کدام است؟ $\mathbf{T(n)=4T(\frac{n}{2})+n^2\log n}$

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

۱۰ـ از بین موارد مقابل کدام مورد صحیح است؟ الف) $\mathbf{n^n \in O(n!)}$ ب) $\mathbf{2^n \in \Omega(n^{100})}$ ج) $\mathbf{(1.01)^n \in O(n^3)}$

۱) الف و ب
۲) ب و ج
۳) فقط ج
۴) فقط ب

۱۱ـ مرتبه زمانی رابطه مقابل کدام است؟ $\mathbf{T(n)=T(\frac{2n}{3})+\log^2 n}$

۱) $\theta(n\log n)$
۲) $\theta(n^{\log_2^3})$
۳) $\theta(\log^3 n)$
۴) $\theta(n)$

۱۲ـ مرتبه زمانی رابطه مقابل چیست؟ $\mathbf{T(n)=1000T(\frac{n}{999})+n}$

۱) $\theta(n)$
۲) $\theta(n^2)$
۳) $\theta(n^{\log_{999}^{1000}})$
۴) $\theta(n\log n)$

۱۳ـ از بین موارد زیر چند مورد نادرست است؟

الف) $\Omega(f(n)) \subseteq \theta(f(n))$ ب) $\Omega(f(n))+\theta(f(n))=\theta(f(n))$ ج) $\Omega(f(n))+O(f(n))=\Omega(f(n))$

۱) یک مورد
۲) دو مورد
۳) سه مورد
۴) همگی درست می‌باشند.

۱۴ـ مرتبه زمانی تابع مقابل کدام است؟ $T(n)=16T(\frac{n}{4})+n^2\log n$

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

۱۵ـ مرتبه زمانی قطعه برنامه زیر چیست؟

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

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

۱۶ـ مرتبه زمانی تابع مقابل کدام است؟ $T(n)=2T(\frac{n}{2})+\frac{n}{\log n}$

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

۱۷ـ مرتبه زمانی قطعه برنامه زیر کدام است؟

```
for (i = 1 ;i < n ;i/ = 5)
     for (j = n ; j > 1; j*= 3)
          for (k = n;k > 1 ;k* = 2)
               x++;
```

۱) $\log n$

۲) $\log^2 n$

۳) $n\log n$

۴) $\log^3 n$

۱۸ـ مرتبه زمانی مربوط به تابع مقابل کدام است؟ $T(n)=T(n-1)+\frac{4}{n}$

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

۱۹ـ مرتبه $\sum_{i=1}^{n} i^k$ کدام است؟

۱) $\theta(n^{k+1})$ ۲) $\theta(k^n)$ ۳) $\theta(n^2)$ ۴) $\theta(n^3)$

۲۰ـ مرتبه رشد تابع $T(n)=2T(\frac{n}{4})+\sqrt{n}$ کدام است؟

۱) $n\log n$ ۲) $n\log^2 n$ ۳) $\sqrt{n}\log\sqrt{n}$ ۴) $n^2\log n$

۲۱ـ کار تابع f بر روی رشتهٔ s با n کاراکتر چیست؟

(تابع sub(s,i,n) تعداد n کاراکتر از موقعیت i در رشته s را برمی‌گرداند.)

$$f(s,n)=\begin{cases} S & \text{اگر } n=1 \\ Sub(s,n,1)+f(sub(s,1,n-1),n-1) & \text{اگر } n>1 \end{cases}$$

۱) رشتهٔ s را برمی‌گرداند.
۲) معکوس رشتهٔ s را برمی‌گرداند.
۳) یک کاراکتر از انتها به رشتهٔ s اضافه می‌کند.
۴) یک کاراکتر به ابتدای رشتهٔ s اضافه می‌کند.

۲۲ـ مجموع مراحل خطوط در برنامهٔ زیر چند است؟

```
int  f(int  x[],int  L)
{
int  i,t = 0;
for(i = 0;i < L;i++)
        {t+ = x[i];
         x[i] = 0;
         }
            return   t;
}
```

۱) 2L+3
۲) 2L+1
۳) 3L+3
۴) 3L+2

۲۳ـ برای یافتن داده‌ای در یک لیست مرتب شده شامل 5000 عنصر حداکثر چند مقایسه مورد نیاز است؟ (روش جستجوی دودویی)

۱) 10 ۲) 25 ۳) 30 ۴) 13

۲۴ـ در برنامهٔ زیر مقدار F(4,5) برابر کدام است ؟

```
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) 8
۲) 10
۳) 20
۴) هیچ‌کدام

۲۵ـ با توجه به تابع زیر مقدار func(n) چه خواهد بود؟

```
int func (int n)
{  if (n= =0 )return (0) ;
return (n+func (n-1)) ;
}
```

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

۲۶ـ مجموع مراحل خطوط در تکه برنامه زیر چند است؟

```
for (i=1;i<=n;++i)
for(j=1;j<=n;++j)
{
c[i][j]=0;
for(k=1;k<=n;++k)
c[i][j]+=a[i][k]*b[k][j];
   }
```

۱) $2n^3+2n^2+n+1$
۲) $2n^3+2n^2+2n+1$
۳) $2n^3+3n^2+2n+1$
۴) هیچ‌کدام

۲۷ـ مقدار bino(4,2) که از الگوریتم زیر محاسبه می‌شود چیست؟

```
function   bino(n,m:integer):integer;
   begin
     if  n = o or n = m then
        bino: =1
     else
       bino: = bino (n-1, m)+bino(n-1, m-1)
   end;
```

۱) 13
۲) 11
۳) 12
۴) 10

۲۸ـ اگر M واحد زمان صرف اجرای Modul A شود و N تعداد داده‌های ورودی باشد تابع پیچیدگی زمانی الگوریتم زیر کدام است؟

```
I := M
while   I > 1
for   j := 1  to  N
Modul A
end for
I := I / 5
end while
```

۱) $MN\log_5 N$
۲) $MN\log_2 N$
۳) $N\log_5^M$
۴) $MN\log_5 M$

۲۹ـ شمارش گام‌های برنامهٔ زیر کدام است؟

```
void   main( )
{int  i=1  ,  s=0;
while(i<=n)
{i=2*i;
  S+=i;
  }
}
```

۱) $3([\log_2^n]+1)$
۲) $3([\log_2^n]+2)$
۳) $3[\log_2^n]+4$
۴) $3([\log_2^n]+1)+2$

۳۰ـ مقدار تابع $a(4,3)$ کدام است؟

$$a(n,m)=\begin{cases}1 & \text{اگر } n=o \text{ یا } n=m\\ a(n-1,m)+a(n-1,m-1) & \text{در غیر این صورت}\end{cases}$$

۱) 3 ۲) 5 ۳) 4 ۴) 6

۳۱ـ کدام گزینه نادرست می‌باشد؟

۱) $4^{\log n}\in\Omega(\log^2 n)$ ۲) $4^{\log n}\in O(\log n^{\log n})$ ۳) $\log^2 n\in O(\log n^{\log n})$ ۴) $\log^2 n\in\Omega(4^{\log n})$

۳۲ـ اگر $g(n)\in O(h(n)), f(n)\in\theta(h(n))$ کدام گزینه نادرست است؟

۱) $g(n)+f(n)\in\Omega(h(n))$ ۲) $g(n)+f(n)\in O(h(n))$ ۳) $h(n)\in\Omega(g(n))$ ۴) $g(n)\in\Omega(h(n))$

۳۳ـ فرض کنید g,f دو تابع دلخواه به شکل $f,g:N\to R^+$ باشند، به علاوه فرض کنید $\lim\limits_{n\to\infty}\frac{f(n)}{g(n)}=\circ$. کدام‌یک از گزینه‌های زیر درست است؟

۱) $g(n)\in O(f(n)), g(n)\in\Omega(f(n))$ ۲) $f(n)\in O(g(n)), f(n)\notin\Omega(g(n))$

۳) $f(n)\in O(g(n)), f(n)\in\Omega(g(n))$ ۴) $f(n)\notin O(g(n)), f(n)\in\Omega(g(n))$

۳۴ـ مقدار $f(52,16,16)$ را با توجه به قطعه کد زیر به دست آورید.

```
int f(int a, int b, int c)
{if (b == 0)return(1);
else if (a%b >= c)return(c);
     else return(c*b*f(a/2, b/2, c/2));
}
```

۱) 1048576
۲) 4096
۳) 1
۴) 65536

۳۵ـ رابطه غیر بازگشتی برای تابع بازگشتی زیر کدام است؟

$$f(x)=\begin{cases}2f(\frac{x}{2})+3x-2 & \text{اگر } x>1\\ 1 & \text{اگر } x=1\end{cases}$$

۱) $3x\log_2^x-3x+1$ ۲) $3x\log_2^x+x-2$

۳) $3x\log_2^x-x+2$ ۴) $3x\log_2^x-3x+3$

۳۶ـ پیچیدگی زمانی تابع بازگشتی مقابل کدام است؟

```
int f(int n)
{if (n <= 1)return(2);
     else return(f(n/4)+f(n/4)*f(n/4));
}
```

۱) $O(n^{\log_3^4})$ ۲) $O(n^{\log_4^3})$

۳) $O(n^{\frac{4}{3}})$ ۴) $O(n^{\frac{3}{4}})$

۳۷ـ مرتبه $T(n)$ کدام است؟

$$T(n)=5T(\frac{n}{3})+8n$$

۱) $\theta(n)$ ۲) $\theta(n\log n)$

۳) $\theta(n^{\log_5^3})$ ۴) $\theta(n^{\log_3^5})$

۳۸ـ مرتبه $T(n)$ کدام است؟

$$T(n)=27T(\frac{n}{3})+2n^3$$

۱) $\theta(n^3\log n)$ ۲) $\theta(n^{\log_2^{27}})$ ۳) $\theta(n^3)$ ۴) $\theta(n^2)$

۳۹ـ مرتبه‌ی $T(n)$ کدام است؟

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

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

۴۰ـ مرتبه‌ی $T(n)$ کدام است؟

$$T(n)=3T(n-2)+4$$

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

فصل دوم

«آنالیز سرشکن»

تست‌های تألیفی فصل دوم

مثال۱: n قطعه شکلات با طول‌های $\{2,4,6,...,2n\}$ در اختیار داریم و می‌خواهیم آن‌ها را خرد نماییم. اگر طول قطعه شکلات فرد باشد، هنگام خم شدن، خرد می‌شود ولی اگر طول آن زوج باشد، نیمی از شکلات سالم می‌ماند و نیمه دیگر آن خرد می‌شود. اگر k عمل خم کردن تکه شکلات‌ها نیاز باشد تا همه شکلات‌ها خرد شوند، مقدار k از چه مرتبه‌ای خواهد بود؟

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

☑ **پاسخ:** گزینه «۱» هر تکه شکلات با طول $c2^n$ (که c عددی فرد است) با n+1 خم شدن، بطور کامل خرد خواهد شد. در این صورت اگر n را برابر $2^{m-1}-1$ درنظر بگیریم، 1 تکه شکلات به m خم شدن، 2 تکه شکلات به m-1 خم شدن، 4 تکه شکلات به m-2 خم شدن، .... ، 2^k تکه شکلات به n-k خم شدن، ... و 2^{m-2} تکه شکلات به 2 خم شدن نیاز دارند. مجموع این مقادیر از مرتبه $\theta(n)$ خواهد بود.

مثال۲: الگوریتمی داریم که زمان اجرایش روی ورودی عدد مربع کامل i برابر i و روی ورودی سایر اعداد برابر 1 می‌باشد. میانگین هزینه اجرای این الگوریتم روی اعداد مجموعه $\{1,2,...,n\}$ از چه مرتبه‌ای خواهد بود؟

۱) $\theta(1)$ ۲) $\theta(\ln n)$ ۳) $\theta(\sqrt{n})$ ۴) $\theta(n)$

☑ **پاسخ:** گزینه «۳» برای هر مقدار n می‌توان مقدار m یافت بطوری که در رابطه $m^2 \le n < (m+1)^2$. در این صورت در بازه اعداد 1 تا n، به تعداد m عدد مربع کامل و n-m عدد غیرمربع کامل داریم. میانگین زمان اجرای این الگوریتم روی این داده‌ها برابر است با:

$$\text{mean} = \frac{\sum_{i=1}^{n-m} 1 + \sum_{i=1}^{m} i^2}{n} = \frac{n-m+\frac{m(m+1)(2m+1)}{6}}{n} \cong \frac{m^3}{3m^2} \in \theta(m) = \theta(\sqrt{n})$$

مثال۳: یک رجیستر 2n بیتی در اختیار داریم و قصد داریم تمام 2^{2n} عدد قابل ذخیره در این رجیستر را پشت سر هم در آن ذخیره نماییم. در صورتی که هزینه ثبت هر عدد در این رجیستر برابر با تعداد بیت‌های متفاوت عدد پیشین و عدد فعلی باشد، هزینه سرشکن شده هر ذخیره‌سازی در بدترین شرایط (بدترین ترتیب از 2^{2n} عدد) چه مقدار است؟

۱) $\frac{3n}{4}$ ۲) $\frac{3n}{2}$ ۳) $2n-1$ ۴) n

☑ **پاسخ:** گزینه «۳» رشته‌های 2n بیتی از اعداد را می‌توان به نحوی مرتب نمود که تعداد بیت‌های متمایز هر دو عدد متوالی در این ترتیب برابر 1 باشد. این ترتیب به کدگذاری گری معروف است. با توجه به اینکه طول هر رشته برابر 2n بیت است و در این ترتیب، هر دو عدد متوالی دقیقاً در 1 بیت اختلاف دارند، می‌توان نتیجه گرفت که اعداد با تعداد زوج (یا فرد) بیت با مقدار 1، نسبت به هم فاصله زوج دارند. برای تمام اعداد در جایگاه‌های فرد از این ترتیب، مقدار بیت‌ها را قرینه می‌کنیم. (بجای مقدار 0 مقدار 1 قرار می‌دهیم و بالعکس) با این کار، فقط ترتیب جایگاه اعداد در جایگاه فرد عوض می‌شود و در این دنباله، دو عدد مشابه نخواهیم داشت. در ترتیب جدید، تعداد بیت‌های مشابه هر دو عدد متوالی برابر 1 و تعداد بیت‌های متفاوت برابر 2n-1 خواهد بود. در نتیجه هزینه هر ذخیره‌سازی برابر 2n-1 است.

مثال۴: تابعی را روی اعداد $\{1,2,3,...,n\}$ اجرا می‌کنیم. درصورتی که پیچیدگی محاسباتی این تابع روی همه اعداد مانند i بجز اعداد مکعب کامل از مرتبه $\theta(n)$ و پیچیدگی محاسباتی میانگین آن بر روی همه اعداد از مرتبه $\theta(n\lg n)$ باشد، اگر پیچیدگی محاسباتی این تابع برای اعداد مکعب کامل مانند u از مرتبه $\theta(f(n))$ باشد، رابطه $f(n)$ به چه شکل خواهد بود؟

۱) $\theta(n^2\log n)$ ۲) $\theta(n^2\sqrt{n})$ ۳) $\theta(n\sqrt{n}\log n)$ ۴) $\theta(\frac{n^2}{\log n})$

☑ **پاسخ:** گزینه «۳» تعداد اعداد مکعب کامل برابر است با $[\sqrt{n}]$. حال هزینه میانگین اعمال تابع روی تمام اعداد را محاسبه می‌نماییم.

$$T(n) = \frac{(n-\sqrt{n})\times n + \sqrt{n}\times f(n)}{n} \in \theta(n\log n) \Rightarrow \frac{f(n)}{\sqrt{n}} \in \theta(n\log n) \Rightarrow f(n) \in \theta(n\sqrt{n}\log n)$$

آزمون فصل دوم

۱ـ هزینه سرشکن شده تعداد بیت‌های تغییرکرده در k بار اجرای عمل افزایش بر روی یک شمارنده دودویی k بیتی کدام است؟ (هزینه تغییر هر بیت $O(1)$ است.)

۱) $O(1)$		۲) $O(3)$		۳) $O(n)$		۴) $O(\log_2^n)$

۲ـ در صورتی‌که 2^k مرتبه مجاز به افزایش یا کاهش این شمارنده با ترتیب دلخواه باشیم، هزینه سرشکن شده از چه مرتبه‌ای خواهد بود؟

۱) $O(1)$		۲) $O(\log k)$		۳) $O(k)$		۴) $O(2)^k$

۳ـ اگر 2^k مرتبه با ترتیب دلخواه مجاز به افزایش یا کاهش مقدار شمارنده باشیم و عمل کاهش هزینه‌ای نداشته باشد (فقط عمل افزایش هزینه‌ای برابر با تعداد بیت‌های تغییرکرده داشته باشد) هزینه سرشکن این n عمل به چه صورت است؟

۱) $O(1)$		۲) $O(\log k)$		۳) $O(k)$		۴) $O(2)^k$

۴ـ اگر تنها 2^k مرتبه عمل افزایش را انجام دهیم و هزینه تغییر بیت iام برابر i باشد، هزینه سرشکن شده این 2^k عمل از چه مرتبه‌ای خواهد بود؟

۱) $O(1)$		۲) $O(\log k)$		۳) $O(\log\log k)$		۴) $O(k)$

۵ـ اگر تنها 2^k مرتبه عمل افزایش را انجام دهیم و هزینه تغییر بیت iام برابر 2^i باشد، هزینه سرشکن شده این 2^k عمل از چه مرتبه‌ای خواهد بود؟

۱) $O(1)$		۲) $O(\log k)$		۳) $O(\log\log k)$		۴) $O(k)$

فصل سوم

«مقدمه‌ای بر پیچیدگی محاسباتی (Computational Complexity)»

آزمون فصل سوم

۱ـ کدام‌یک از موارد زیر قطعاً برقرار است؟

۱) $P \subseteq NP$ ۲) $P = NP$ ۳) $P \neq NP$ ۴) $NP \subseteq P$

۲ـ اگر یک مسئله تصمیم‌گیری مانند A در زمان چندجمله‌ای قابل حل باشد، آنگاه از بین موارد زیر چند مورد صحیح است؟

الف) $A \in P$ ب) $A \in NP$ ج) $A \in NPC$

۱) یک مورد ۲) دو مورد ۳) سه مورد ۴) هیچ‌کدام

۳ـ دو مورد زیر را در نظر بگیرید:

الف) $CO-NP = NP$ ب) $CO-NP \cap NP \neq \varnothing$

کدام‌یک از موارد بیان شده با توجه به دانش امروزی صحیح می‌باشد؟

۱) فقط الف ۲) فقط ب ۳) الف و ب ۴) هیچ‌کدام

۴ـ با توجه به دانش امروزی، کدام مورد اثبات یا رد نشده است؟

۱) $NP \cap NPC \neq \varnothing$ ۲) $NP-hard \cap NP \neq \varnothing$ ۳) $P \cap NP \neq \varnothing$ ۴) $CO-NP = NP$

۵ـ اگر A یک مسئله NP– کامل باشد کدام‌یک از موارد زیر قطعاً در مورد این مسئله صحیح است؟

۱) A در زمان چند جمله‌ای قابل حل است.
۲) A در زمان چند جمله‌ای قابل حل نیست.
۳) A یک مسئله NP– سخت است.
۴) A یک مسئله NP است.

۶ـ اگر مسئله ماکزیمم کلیک در زمان $O(|E|+|V|)$ قابل حل باشد، آنگاه می‌توان گفت:

۱) $P = NP$
۲) $P \subseteq NP$
۳) $P \cap NP \neq \varnothing$
۴) هیچ اطلاعاتی راجع به رابطه بین P و NP ایجاد نمی‌شود.

۷ـ کدام‌یک از مسائل زیر NP– سخت نمی‌باشند؟

۱) یافتن دور هامیلتونی با کمترین وزن در یک گراف وزن‌دار
۲) مسئله پوشش رأسی (Vertex Cover)
۳) مسئله یافتن کوتاه‌ترین مسیرها بین تمام جفت رئوس در یک گراف وزن‌دار
۴) مسئله یافتن طولانی‌ترین مسیرها بین تمام جفت رئوس در یک گراف وزن‌دار

۸ـ فرض کنید A یک مسئله NP– کامل باشد و داشته باشیم $A \propto_P B$ در این صورت در مورد B می‌توان گفت (داریم $B \in NP$):

۱) یک مسئله P است.
۲) یک مسئله NP– سخت است.
۳) یک مسئله NP– کامل است.
۴) ممکن است یک مسئله NP– کامل باشد.

۹ـ کنترل درست یا نادرست بودن یک حل کاندید در مسئله دور هامیلتونی دارای چه زمانی است؟

۱) $O(n)$ ۲) $O(n!)$ ۳) $O(n^n)$ ۴) $O(2^n)$

۱۰ـ اگر A یک مسئله NP– سخت باشد، آنگاه از بین موارد زیر کدام‌یک در مورد A صحیح است؟

الف) حداقل به سختی مسائل NP– کامل است.
ب) قطعاً در زمان چند جمله‌ای قابل حل نیست.

۱) فقط الف ۲) فقط ب ۳) الف و ب ۴) هیچ‌کدام

بخش دوم: داده‌ساختارها

فصل چهارم

«داده‌ساختارهای مقدماتی»

تست‌های تألیفی فصل چهارم

مثال ۱: آرایهٔ A را در پاسکال به صورت زیر تعریف کرده‌ایم. اگر آرایه در آدرس 1000 حافظه قرار داشته باشد و اندازه عنصر integer برابر 2 بایت در نظر گرفته شود، آدرس عنصر A[14] به روش سطری کدام است؟

$A = \text{array}[3..50] \text{ of integer;}$

۱) 1024 ۲) 1028 ۳) 1014 ۴) 1022

پاسخ: گزینه «۴» با استفاده از فرمول ارائه شده برای آرایه یک بعدی داریم:

$$1000 + (14-3)\times 2 = 1000 + 11\times 2 = 1022$$

مثال ۲: $k[4,2,5]$ چندمین عنصر به روش ستونی از آرایهٔ تعریف شدهٔ روبرو محسوب می‌شود؟ $k = \text{array}[-1..6, 0..5, 2..6]$

۱) 165 ۲) 167 ۳) 166 ۴) 164

پاسخ: گزینه «۳» با استفاده از فرمول آرایه سه بعدی می‌توان نوشت:

$$(5-2)\times 6\times 8 + (2-0)\times 8 + (4-(-1)+1) = 3\times 48 + 2\times 8 + 6 = 144 + 16 + 6 = 166$$

مثال ۳: فرض کنید آرایه‌ای سه‌بعدی در اختیار داریم که داده‌ها در یک هرم (مشابه مسأله ماتریس پایین مثلثی) با ترتیب زیر ذخیره شده‌اند:

(1,1,1),(2,1,1),(2,1,2),(2,2,1),(3,1,1),(3,1,2),(3,1,3),(3,2,1),(3,2,2),(3,3,1),(4,1,1),...

یعنی داده‌ها صفحه به صفحه از بالا به پایین و در هر صفحه مانند صفحه kام، داده‌ها روی مثلث به‌صورت سطر به سطر از عنصر (k,1,1) تا عنصر (k,1,k) و پس از آن از (k,2, 1) تا عنصر (k,2,k-1) و به همین ترتیب، ذخیره می‌شوند. در این صورت مختصات عنصر 100ام در این ترتیب چه خواهد بود؟

۱) (9,2,3) ۲) (8,3,1) ۳) (9,1,5) ۴) (8,4,3)

پاسخ: گزینه «۲» با توجه به اطلاعات مسأله، عناصر صفحه iام مقادیری از (i,j,k) هستند که در رابطه $j+k \le i+1$ صدق می‌کند و مقادیر j و k مقادیر طبیعی هستند. در این صورت صفحه iام از این هرم به تعداد $\frac{i(i+1)}{2}$ عنصر دارد. با محاسبه مقدار $\sum_{j=1}^{i}\frac{j(j+1)}{2}$ می‌توان نتیجه گرفت شماره عناصر موجود در صفحه iام از $+1$ $\frac{(i-1)i(i+1)}{6}$ تا $\frac{i(i+1)(i+2)}{6}$ می‌باشد. عناصر شماره 85 تا 120 در صفحه هشتم قرار خواهند داشت. جواب مسأله برابر با مختصات صدمین عنصر هرم که شانزدهمین عنصر از این صفحه است خواهد بود. مختصات این عنصر به‌صورت (8,3,1) می‌باشد. (مختصات اولین عنصر این صفحه (8,1,1)، هشتمین عنصر صفحه (8,1,8) و پانزدهمین عنصر صفحه (8,2,7) است.)

مثال ۴: برای حذف‌کردن یک عنصر از یک لیست پیوندی دو طرفه چند آدرس باید جایگزین شود؟

۱) چهار ۲) سه ۳) دو ۴) یک

پاسخ: گزینه «۳» برای حذف‌کردن یک عنصر از یک لیست پیوندی دو طرفه باید دو آدرس جایگزین شود. به مطالب ارائه شده در این بخش مربوط به لیست‌های پیوندی دو طرفه مراجعه کنید.

مثال ۵: معادل میان‌وندی (infix) عبارت پسوندی (postfix) روبرو چیست؟ $ABC-{}^\wedge D/E+$

۱) $A^\wedge B-C/D+E$ ۲) $A^\wedge(B-C)/D+E$ ۳) $A^\wedge B-C/(D+E)$ ۴) $A^\wedge(B-C)/(D+E)$

پاسخ: گزینه «۲» با استفاده از روش پرانتزگذاری داریم:

$$(((A(BC-)^\wedge)D/)E+) = (((A(B-C)^\wedge)D/)E+) = (((A^\wedge(B-C))D/)E+)$$
$$= (((A^\wedge(B-C))/D)E+) = (((A^\wedge(B-C))/D)+E) = A^\wedge(B-C)/D+E$$

مثال ۶: معادل پسوندی (Postfix) عبارت ریاضی میان‌وندی (Infix) روبرو کدام است؟ $x+y/(u-t*w)*y$

۱) $xyutwy*-*/+$ ۲) $xyutw*-/y*+$ ۳) $utw*-xy/+y*$ ۴) $tw*u-y/x+y*$

☑ **پاسخ:** گزینه «۲» ابتدا پرانتزگذاری کرده و سپس عملگرها را به پرانتز بسته‌ها منتقل می‌کنیم:

$$(x+((y/(u-(t*w)))*y)) \Rightarrow (x+((y/(u-(tw*)))*y)) \Rightarrow (x+((y/(utw*-))*y))$$
$$\Rightarrow (x+((yutw*-/)*y)) \Rightarrow (x+(yutw*-/y*)) \Rightarrow xyutw*-/y*+$$

مثال ۷: معادل پسوندی عبارت $\left(A-B/(C*D\,\hat{}\,E)\right)$ کدام است؟

۱) $ABCD*E\,\hat{}\,-/$ ۲) $ABCDE\,\hat{}\,*/-$ ۳) $AB-CD*E\,\hat{}\,/$ ۴) $AB/CDE\,\hat{}\,*-$

☑ **پاسخ:** گزینه «۲» با استفاده از روش پرانتزگذاری داریم:

$$(A-(B/(C*(D\,\hat{}\,E)))) \Rightarrow (A-(B/(C*(DE\hat{})))) \Rightarrow (A-(B/(CDE\,\hat{}\,*))) \Rightarrow (A-(BCDE\,\hat{}\,*/)) \Rightarrow ABCDE\,\hat{}\,*/-$$

مثال ۸: پشته (STACK) ساختمان داده‌ای است از نوع:

۱) IFOF ۲) LIFO ۳) FIFO ۴) LFIO

☑ **پاسخ:** گزینه «۲» در پشته اولین ورودی آخرین خروجی است یا Last In First out.

مثال ۹: POP و PUSH کردن به ترتیب به چه معناست؟

۱) برداشتن عنصر بالایی پشته ـ گذاشتن عنصر جدید روی پشته
۲) خالی کردن پشته ـ پر کردن پشته
۳) دیدن عنصر بالایی پشته ـ برداشتن عنصر بالایی پشته
۴) گذاشتن عنصر در پشته ـ خواندن عنصر بالایی پشته

☑ **پاسخ:** گزینه «۱» POP، به معنی حذف عنصر بالایی پشته و PUSH، به معنی درج عنصری روی پشته است.

آزمون فصل چهارم

۱ـ آرایهٔ دو بعدی $A[-1..7,0..10]$ در آدرس **400** به بعد حافظه قرار دارد و هر خانه آرایه احتیاج به **4** بایت دارد. آدرس عنصر $A[4,10]$ به روش ستونی کدام است؟

۱) 766 ۲) 810 ۳) 860 ۴) 780

۲ـ اگر عناصر غیر صفر ماتریس بالا مثلثی $A_{n\times n}$ را در آرایهٔ خطی **B** ذخیره کنیم فرمولی که بر اساس آن اندیس **k** درآرایهٔ **B** برای یافتن عنصر $A(i,j)$ به دست می‌آید، کدام است؟

۱) $k=2i+j-2$

۲) $k=ni+\frac{1}{2}i-\frac{1}{2}i^2-n+j$

۳) $k=\frac{1}{2}i^2-\frac{1}{2}i+j$

۴) $k=\frac{1}{2}i^2-\frac{1}{2}i+j-1$

۳ـ $A[3,6,-1]$ چندمین عنصر از آرایهٔ تعریف شدهٔ $A=array[2..5,3..8,-5..0]$ محسوب می‌شود؟

۱) 58 ۲) 60 ۳) 59 ۴) 53

۴ـ کدام یک از روال‌های زیر عنصر جدید را بعد از عنصر **x** اضافه می‌کند؟

۱)
```
void insert (pointer x,pointer y)
{
   y→ Link = x→Link;y→Link = x;
}
```

۲)
```
void insert (pointer x , pointer y)
{
   x→ Link = y → Link ; y →Link = x;
}
```

۳)
```
void insert (pointer x , pointer y)
{
   y→ Link = x →Link ; x→ Link = y ;
{
```

۴)
```
void insert (pointer x , pointer y)
{
   y→Link = x →Link ; x= y ;
}
```

۵ـ کدام یک از موارد زیر از لحاظ پیچیدگی عملیات بین لیست یک طرفه و دو طرفه فرقی نمی‌کند؟

۱) پیمایش لیست از انتها به ابتدا
۲) پیمایش لیست از ابتدا به انتها
۳) حذف گره با آدرس p از لیست
۴) درج گره در انتهای لیست

۶ـ عملکرد الگوریتم زیر چیست؟

```
List k (List L , int b)
{List a ;
 a = L →Link ;
 while (a!= NULL && a → data!=b)
 a = a → Link ;
 return a ;
 }
```

۱) پیدا کردن مکان عنصر b
۲) پیمایش لیست از انتها به ابتدا
۳) پیدا کردن مکان نود بعد از عنصر b
۴) پیدا کردن محل عنصر b در لیست دو طرفه

۷ـ عملکرد روال زیر در یک لیست دو طرفه کدام است؟

```
void k (dpointer p , dpointer x )
{
Llink (RLink (p))= x;
RLink (x)= RLink (p);
RLink (p) = x;
Llink (x) = p;
}
```

۱) اضافه کردن گرهٔ x به سمت راست گرهٔ p .
۲) اضافه کردن گرهٔ p به سمت راست گرهٔ x .
۳) اضافه کردن گرهٔ x به سمت چپ گرهٔ p .
۴) اضافه کردن گرهٔ x به سمت راست گرهٔ بعد از P .

۸ـ با توجه به صف حلقوی زیر چنانچه عناصر **p** و **q** و **r** به ترتیب از سر چپ آن حذف و عناصر **x** و **y** و **z** به ترتیب به سر راست آن اضافه شوند، مقدار **Left** و **Right** به ترتیب از راست به چپ چه خواهد بود؟

Left:5
Right : 9

1	2	3	4	5	6	7	8	9	10
				p	q	r	s	t	

۱) $1-8$ ۲) $3-8$ ۳) $2-8$ ۴) $10-9$

۹ـ عبارت پیشوندی $-- + cba \uparrow - ed + c / ab$ داده شده است، معادل پسوندی آن کدام است؟

۱) $cb + a - ed - c + ab / \uparrow -$

۲) $cb + a - ed - cab / + \uparrow -$

۳) $cb + a - ed - cab + / \uparrow -$

۴) $cb + a - ed - c \uparrow ab / + -$

۱۰ـ اگر اعداد $5, 4, 3, 2, 1$ را به ترتیب به یک پشته وارد کنیم، کدام یک از خروجی‌های زیر از این پشته امکان‌پذیر نخواهد بود؟ (خروجی‌ها را از سمت چپ به راست بخوانید)

۱) $5-4-3-2-1$

۲) $1-2-3-4-5$

۳) $1-3-5-4-2$

۴) $1-3-5-2-4$

فصل پنجم

«داده‌ساختارهای مبتنی بر گراف»

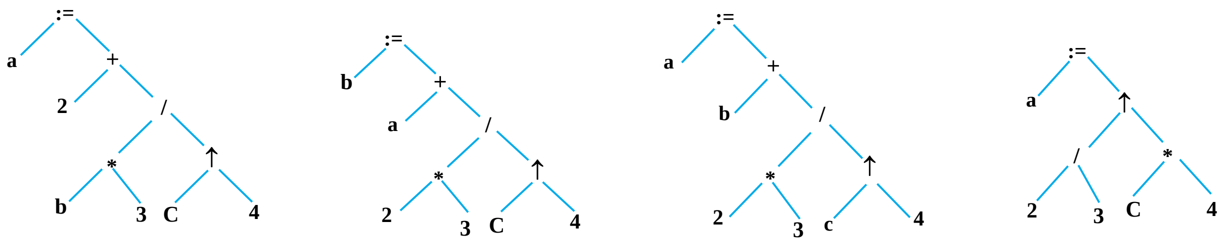
تست‌های تألیفی فصل پنجم

مثال۱: در نمایش و ارزیابی عبارت‌های ریاضی با استفاده از درخت‌های دودویی، کدام یک از روش‌های طی‌کردن درخت فرم طبیعی عبارت را به دست می‌دهند؟

۱) Postorder ۲) Preorder ۳) Levelorder ۴) Inorder

☑ **پاسخ:** گزینه «۴» در فرم طبیعی عبارات ریاضی همواره عملگر (ریشه)، بین عملوندهای خود قرار می‌گیرد.

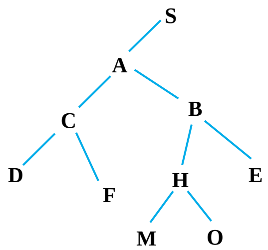
مثال۲: عبارت $:= a + b / *23 \uparrow c4$ پیمایش **Preorder** کدام یک از درخت‌های زیر است؟



(۱) (۲) (۳) (۴)

☑ **پاسخ:** گزینه «۲» در روش پیشوندی یا Preorder ابتدا ریشه و سپس زیر درخت‌های چپ و راست به همین ترتیب پیمایش می‌شوند.

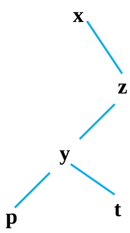
مثال۳: نتیجه پیمایش درخت زیر به روش **LVR** چیست؟



۱) DCFAMHOBES
۲) DFCMOHEBSA
۳) SACDFBHMOE
۴) SABEHOMCFD

☑ **پاسخ:** گزینه «۱» در پیمایش به روش LVR، ابتدا زیر درخت چپ و بعد ریشه و سپس زیر درخت راست را پیمایش می‌کنیم.

مثال۴: پیاده‌سازی درخت زیر با آرایه به کدام صورت صحیح است؟



۱)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x		z			y						p	t		

۲)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x	z	y	p	t										

۳)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
		t	p						y			z		x

۴)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x		z	y	p	t									

☑ **پاسخ:** گزینه «۱» ابتدا گره‌های درخت مربوطه را براساس یک درخت پُر، شماره‌گذاری کرده و سپس با همان شماره‌ها در اندیس‌های آرایه قرار می‌دهیم.

مثال ۵: در کدام پیمایش می‌توان با استفاده از دستور Delete به جای دستور Cout، تمام گره‌های یک درخت دودویی را حذف کرد؟

۱) RLV ۲) RVL ۳) VLR ۴) LVR

☑ **پاسخ:** گزینه «۱» در هر کدام از پیمایش‌ها که ریشه قبل از زیردرخت چپ یا راست پیمایش می‌شود، اگر از دستور Delete استفاده کنیم، آدرس بقیه‌ی درخت گم می‌شود. پس می‌توان دستور Delete را در پیمایش RLV به کار برد که ریشه در آخر ملاقات می‌شود.

مثال ۶: کدام‌یک از گزینه‌های زیر صحیح است؟

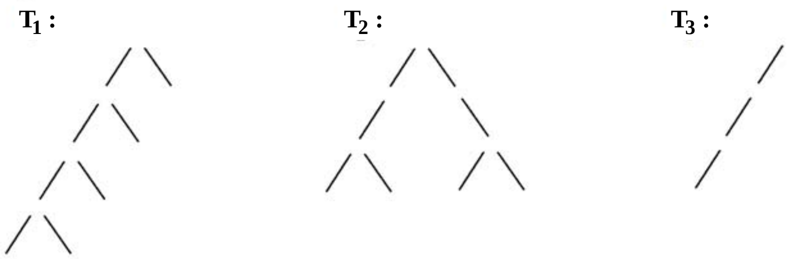
۱) در هر درخت دودویی، اختلاف ارتفاع هر گره و والدش برابر 1 است.

۲) در دو درخت T_1 و T_2 با تعداد گره‌های برابر، همواره درختی که ارتفاع بیشتری دارد تعداد برگ بیشتری نخواهد داشت. (تعداد برگ‌هایش کوچکتر یا مساوی تعداد برگ‌های درخت با ارتفاع کمتر است.)

۳) هر درخت دودویی با 14 گره، حداقل یک گره تک‌فرزندی دارد.

۴) در یک درخت دودویی، تعداد برگ‌ها کمتر از تعداد گره‌های داخلی نخواهد بود.

☑ **پاسخ:** گزینه «۳» عبارت گزینه اول نادرست است. در صورتی که ارتفاع دو فرزند یک درخت برابر نباشد، این رابطه بین فرزند با ارتفاع کمتر و والدش برقرار نیست. عبارت گزینه دوم نیز نادرست است. دو درخت T_1 و T_2 مثال نقضی برای این عبارت هستند. عبارت گزینه (۳) درست است و هر درخت با تعداد زوج گره، حداقل یک گره تک‌فرزندی دارد. زیرا تعداد گره‌های دوفرزندی یک درخت، یکی کمتر از تعداد برگ‌های آن درخت است و مجموع این دو مقدار، عددی فرد خواهد بود. عبارت گزینه (۴) نیز نادرست است. درخت T_3 مثال نقضی برای این گزینه است.



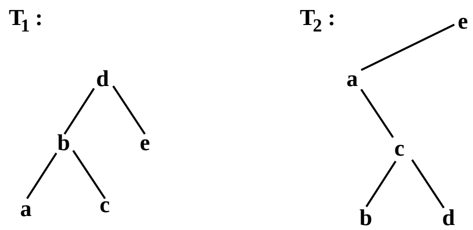
مثال ۷: تعداد درختان دودویی به ارتفاع ۳ با رئوس مجموعه {1,2,3,...,13} برابر با:

۱) $2\times14!$ ۲) $\binom{8}{2}$ ۳) $\binom{7}{2}$ ۴) $\frac{15!}{2}$

☑ **پاسخ:** گزینه «۱» ساختار کلی درختان دودویی با 13 گره با ارتفاع 3 از حذف 2 برگ از درخت دودویی پر با 15 گره بدست می‌آید. درخت پر با 15 گره، 8 برگ دارد و به $\binom{8}{2}$ حالت می‌توان 2 برگ از آن را حذف نمود. با توجه به اینکه جایگاه این رئوس اهمیت ندارد، این 13 رأس به $13!$ حالت قابل برچسب دادن هستند. جواب مسأله برابر است با:

$$\binom{8}{2}\times13!=2\times14!$$

مثال ۸: با توجه به جایگشت‌های مختلف قرار دادن عناصر در درخت جست‌وجوی دودویی، احتمال تشکیل درخت T_1 چند برابر احتمال تشکیل درخت T_2 است؟



۱) 1

۲) 2

۳) 4

۴) 8

☑ **پاسخ:** گزینه «۳» درخت T_1 با 8 جایگشت متفاوت قابل ساخت است. اولین عنصر می‌بایست d باشد. عنصر b نیز می‌بایست قبل از دو عنصر a و c در رشته قرار گیرد. با این فرضیات دو عنصر a و c هرکدام 2 حالت برای ترتیب آمدن دارند و عنصر e در یکی از 4 جایگاه دوم تا پنجم قرار می‌گیرد. درخت T_2 نیز با 2 جایگشت متفاوت تشکیل می‌شود. ترتیب ورودی می‌بایست به یکی از دو حالت e,a,c,b,d یا e,a,c,d,b باشد. در نتیجه احتمال تشکیل درخت T_1 چهار برابر احتمال تشکیل درخت T_2 خواهد بود.

مثال۹: تعداد درختان جستجوی دودویی متمایز با ارتفاع حداقل و با 28 گره برابر است با:

۱) 560 ۲) 568 ۳) 552 ۴) 576

پاسخ: گزینه «۲» کافی است ابتدا تعداد ساختارهای درختان دودویی را محاسبه نماییم و سپس عناصر را در درخت قرار دهیم. با توجه به اینکه این درخت می‌بایست شرط درخت جست‌وجو بودن را داشته باشد، فقط یک جایگشت از عناصر در این درخت مجاز است. تعداد ساختارهای مجاز این درخت برابر است با تعداد راه‌های مجاز حذف 3 گره از درخت پر با 31 گره. این 3 گره می‌توانند 3 برگ از بین 16 برگ این درخت یا یک زیردرخت 3 گره‌ای از بین 8 زیردرخت با عمق 1 درخت باشند. در نتیجه تعداد درختان مجاز برابر خواهد بود با: $\binom{16}{3}+\binom{8}{1}=568$

مثال۱۰: هشتمین کلید در پیمایش inorder یک درخت جستجوی دودویی که پیمایش preorder آن به شکل زیر است (از چپ به راست)، کدام گزینه می‌باشد؟

preorder : 15,10,5,3,1,8,12,14,20,17,16,25,23,28

۱) 8 ۲) 14 ۳) 15 ۴) 16

پاسخ: گزینه «۳» می‌دانیم پیمایش inorder یک درخت جستجوی دودویی، لیست مرتب صعودی کلیدهای درخت را می‌دهد، بنابراین کافیست، کلیدهای ذکر شده در پیمایش preorder درخت را به ترتیب صعودی مرتب کنیم، داریم: inorder :1,3,5,8,10,12,14,15,16,17,20,23,25,28

که از چپ به راست، هشتمین کلید، 15 می‌باشد.

مثال۱۱: چنانچه بخواهیم با استفاده از اعداد 10,20,5,22,7,5,20 به هر ترتیب دلخواه یک درخت جستجوی دودویی بسازیم، آنگاه تعداد درخت‌های جستجوی دودویی که می‌توانیم ایجاد نماییم برابر است با:

۱) 7 ۲) 204 ۳) 429 ۴) 42

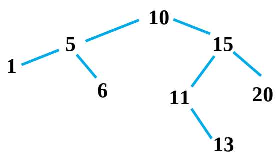
پاسخ: گزینه «۴» در درخت‌های جستجوی دودویی کلید تکراری وجود ندارد، پس مقادیر تکراری در اعداد ذکر شده حذف می‌شود، بنابراین می‌خواهیم تعداد درخت‌های جستجوی دودویی که با 7 مقدار می‌توان ساخت را مشخص کنیم، که برابر با جمله هفتم از اعداد کاتالان است. خواهیم داشت:

$$C_V=\frac{\binom{2n}{n}}{n+1}=\frac{\binom{2\times 5}{5}}{5+1}=\frac{\binom{10}{5}}{6}=\frac{\frac{10!}{5!5!}}{6}=\frac{10\times 9\times 8\times 7\times \cancel{6}\times \cancel{5!}}{5!\cancel{5!}\cancel{6}}=\frac{\cancel{10}\times \cancel{9}^{3}\times \cancel{8}^{2}\times 7}{\cancel{5}\times \cancel{4}\times \cancel{3}\times \cancel{2}}=42$$

مثال۱۲: گره‌های 10,15,5,1,6,20,11,13 را به ترتیب از چپ به راست در یک درخت دودویی جستجوی خالی به نام T درج می‌کنیم. در این صورت پیمایش preorder درخت T کدام است؟ (خروجی‌ها را از چپ به راست بخوانید)

۱) 10,5,1,6,15,11,13,20 ۲) 10,5,1,6,15,13,11,20 ۳) 10,5,6,1,15,11,13,20 ۴) 10,5,6,1,15,13,11,20

پاسخ: گزینه «۱» ابتدا اعداد داده شده را یکی یکی از چپ به راست در یک درخت جستجوی دودویی درج می‌کنیم، داریم:



سپس درخت حاصل را به صورت preorder یا VLR ، یعنی اول ریشه، بعد زیردرخت چپ و در آخر زیردرخت راست را پیمایش می‌کنیم.

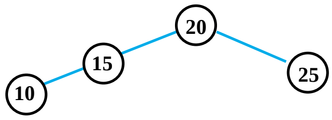
preorder :10,5,1,6,15,11,13,20

مثال۱۳: یک درخت جستجوی دودویی 251 گره دارد. کدام یک از موارد زیر صحیح است؟

۱) این درخت حداقل ۹ سطح و حداکثر 10 سطح دارد.
۲) این درخت حداقل 9 سطح و حداکثر 125 سطح دارد.
۳) این درخت حداقل 8 سطح و حداکثر 9 سطح دارد.
۴) این درخت حداقل 8 سطح و حداکثر 251 سطح دارد.

پاسخ: گزینه «۴» در بدترین حالت یک درخت جستجوی دودویی با 251 گره مورب بوده که در این حالت 251 سطح دارد و در بهترین حالت یک درخت دودویی کامل است که $\left\lfloor Log_2^{251}\right\rfloor+1=7+1=8$ سطح خواهد داشت.

مثال ۱۴: چند حالت عناصر با کلیدهای $10,15,20,25$ را می‌توان وارد یک درخت جستجوی دودویی تهی کرد تا درختی به شکل زیر ایجاد شود؟



۱) 3
۲) 2
۳) 14
۴) 4

☑ **پاسخ:** گزینه «۱» با توجه به شکل درخت جستجوی دودویی داده شده، باید اولین کلید وارد شده 20 باشد که در ریشه قرار گیرد و بعد از 20 می‌توان به یکی از سه حالت زیر بقیهٔ کلیدها را وارد کرد، به طوری که درخت دودویی جستجوی شکل فوق ایجاد شود. تنها محدودیت این است که گره 10 باید بعد از گره 15 به درخت اضافه شود.

$1)20,15,10,25$ $\quad$ $2)20,15,25,10$ $\quad$ $3)20,25,15,10$

مثال ۱۵: چه تعداد از عبارت‌های زیر درست هستند؟

ـ با استفاده از پیمایش پیش‌ترتیب و پس‌ترتیب یک درخت دودویی بدون گره تک‌فرزندی، می‌توان درخت را بطور یکتا بازسازی نمود.
ـ در هر درخت دودویی، اگر m تعداد برگ و n تعداد گره‌های دوفرزندی باشد، همواره رابطه $m=n+1$ برقرار است.
ـ تعداد درختان دودویی قابل ساخت با n گره با مقادیر متمایز، برابر جمله nام از اعداد کاتالان است.

۱) 0 $\quad$ ۲) 1 $\quad$ ۳) 2 $\quad$ ۴) 3

☑ **پاسخ:** گزینه «۳» عبارت اول و دوم درست و عبارت سوم نادرست است. تعداد درختان قابل بازسازی با استفاده از پیمایش پیش‌ترتیب و پس‌ترتیب برابر با 2^k است که k تعداد گره‌های تک‌فرزندی درخت است. تعداد گره‌های دوفرزندی یک درخت، یکی کمتر از تعداد برگ‌های آن درخت است. تعداد درختان دودویی قابل ساخت با n گره برابر با جمله nام از اعداد کاتالان ضربدر $n!$ خواهد بود. زیرا گره‌ها جایگاه مشخصی ندارند و آن‌ها را می‌توان به $n!$ روش جایگشت داد.

مثال ۱۶: درخت ریشه‌داری با ارتفاع 4 در اختیار داریم که تعداد فرزندان گره‌های موجود در ارتفاع i درخت برابر i است. اگر ترتیب پیمایش فرزندان هر گره در این درخت اهمیتی نداشته باشد، چند پیمایش پیش‌ترتیب متمایز از این درخت می‌توانیم داشته باشیم؟

۱) 2^{64} $\quad$ ۲) 2^{40} $\quad$ ۳) $3^4\times2^{19}$ $\quad$ ۴) $3^4\times2^{14}$

☑ **پاسخ:** گزینه «۳» با توجه به اینکه ترتیب فرزندان یک گره اهمیتی ندارد، برای یک گره با i فرزند $i!$ حالت برای ترتیب پیمایش زیردرختان وجود دارد. به‌عنوان مثال، ریشه درخت 4 فرزند دارد و $4!$ حالت برای ترتیب پیمایش زیردرخت‌های آن خواهیم داشت. یعنی پس از پیمایش ریشه، این 4 زیر درخت پشت سر هم پیمایش می‌شوند که تعداد حالات قرار گرفتن پشت سر هم این زیردرختان برابر $4!=24$ است. هر گره 3 فرزندی و 2 فرزندی نیز به‌ترتیب $3!=6$ و $2!=2$ ترتیب برای پیمایش فرزندانشان دارند و گره‌های یک‌فرزندی نیز فقط ۱ حالت برای پیمایش خودشان و تنها فرزندشان خواهند داشت. 1 گره 4 فرزندی، 4 گره 3 فرزندی و 12 گره 2 فرزندی داریم. درنتیجه تعداد کل پیمایش‌ها برابر است با:

$$(4!)^1(3!)^4(2!)^{12}=3^4\times2^{3+4+12}=3^3\times2^{19}$$

مثال ۱۷: یک لیست 9 عنصری حاوی کلیدهای یک تا نه به صورت صعودی مرتب است. اگر این لیست به صورت درجا تبدیل به یک Max heap شود، عنصر چهارم لیست کدام گزینه است؟

۱) 4 $\quad$ ۲) 5 $\quad$ ۳) 6 $\quad$ ۴) 7

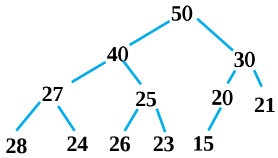
☑ **پاسخ:** گزینه «۱» برای آنکه لیست فوق تبدیل به یک Max heap شود، باید در ابتدا بزرگ‌ترین عنصر در اولین اندیس قرار بگیرد؛ یعنی باید اندیس 9 را با اجدادش، یعنی $[\frac{9}{2}]=4$ ،$[\frac{4}{2}]=2$ ،$[\frac{2}{2}]=1$ جابجا کنیم که نتیجه به صورت: $9,1,3,2,5,6,7,8,4$ در می‌آید، سپس اندیس 8 را با اجدادش، یعنی $[\frac{8}{2}]=4$ ،$[\frac{4}{2}]=2$ جابه‌جا می‌کنیم که نتیجه به صورت: $9,8,3,1,5,6,7,2,4$ در می‌آید. اندیس 7 را با پدرش، یعنی $[\frac{7}{2}]=3$ جابجا می‌کنیم، در نتیجه داریم: $9,8,7,1,5,6,3,2,4$ و اندیس $5,6$ نیاز به جابجایی ندارد و آخرین جابجایی مربوط به مقدار جدید اندیس 9، یعنی عدد 4 و مقدار جدید اندیس 4، یعنی عدد 1 می‌باشد بنابراین داریم: $9,8,7,4,5,6,3,2,1$ که در این لیست عنصر چهارم برابر عدد 4 است.

مثال۱۸: اگر دنباله‌ی زیر بخواهد یک max-heap را نشان دهد کدام اعداد باید با چه مقادیری تعویض شوند؟

(x به y تغییر نماید یعنی $x \rightarrow y$) $50, 40, 30, 27, 25, 20, 21, 28, 24, 26, 23, 15$

۱) $20 \rightarrow 22, 15 \rightarrow 18$　　۲) $28 \rightarrow 31, 25 \rightarrow 27$　　۳) $27 \rightarrow 31, 26 \rightarrow 30$　　۴) $27 \rightarrow 35, 25 \rightarrow 31$

پاسخ: گزینه «۴» با رسم درخت مورد نظر داریم:

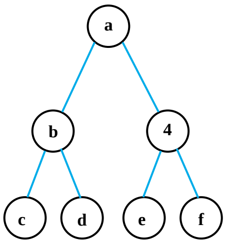


بنابراین باید مقدار 25 به 31 و 27 به 35 تبدیل شود تا این درخت یک max heap را نشان بدهد.

مثال۱۹: با گره‌های {1,2,3,...,7} چند MinHeap می‌توان پیاده‌سازی نمود به طوری که گره 4، سومین عنصر از آرایه معادل با این ساختار باشد؟

۱) 8　　۲) 12　　۳) 16　　۴) 18

پاسخ: گزینه «۲» ساختار این هرم به شکل مقابل خواهد بود.



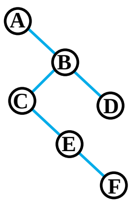
عنصر a فقط می‌تواند مقدار 1 بگیرد. با توجه به اینکه عدد 4 یکی از فرزندان ریشه است، فرزند دیگر ریشه باید عدد 2 باشد. در نتیجه مقدار عنصر b برابر 2 است. عدد 3 باید در یکی از جایگاه‌های c یا d قرار گیرد. در نتیجه 2 حالت دارد. اعداد 5، 6 و 7 می‌توانند در 3 جایگاه باقی‌مانده با !3 حالت قرار می‌گیرد. جواب مسأله برابر است با $2 \times 3! = 12$.

مثال۲۰: درجهٔ یک گره (node) در درخت شامل چیست؟

۱) تعداد فیلدها　　۲) تعداد زیر درخت‌ها　　۳) تعداد ریشه‌ها　　۴) تعداد گره‌ها

پاسخ: گزینه «۲» تعداد زیر درخت‌های یک گره، درجای آن گره است.

مثال۲۱: پیمایش ABCEFD کدام یک از گزینه‌های زیر است؟

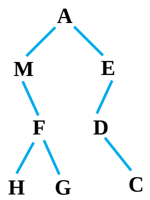


۱) LVR
۲) VLR
۳) VRL
۴) RVL

پاسخ: گزینه «۲» با توجه به پیمایش، چون اول ریشه آمده و بعد ریشهٔ زیر درخت راست آمده باید پیمایشی را انتخاب کنیم که اول ریشه است، و چون بعد از آن، ریشهٔ زیر درخت چپ یعنی C آمده پس باید پیمایش به طریق VLR باشد.

آزمون فصل پنجم

۱ـ نتیجه پیمایش درخت زیر به روش LNR چیست؟



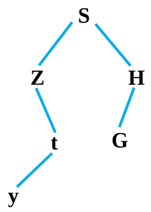
۱) MHFGADCE

۲) HFGMADCE

۳) MHFGACDE

۴) MHFGAEDC

۲ـ پیاده‌سازی درخت زیر با آرایه به کدام صورت صحیح است؟



۱)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S	Z	H		t	G				Y					

۲)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S	Z	H	t	G	y									

۳)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
				y				G	t			H	Z	S

۴)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S	Z	H	t	G	y									

۳ـ پیمایش درخت زیر به روش **postorder** کدام است؟

۱) debfgca

۲) debgfea

۳) edbcgfa

۴) edbgfca

۴ـ پردازهٔ زیر را برای پیمایش درخت دودویی نوشته‌ایم. کدام یک از عبارت‌های زیر درست است؟

```
void trav(tree r)
{
if(r!=NULL){
cout<<r→ info;
trav(r→left);
trav(r→right):
      }
}
```

۱) اگر جای سطرهای ۴ و ۵ پردازه را عوض کنیم، پردازه برای پیمایش پس ترتیبی بکار می‌رود.

۲) این پردازه برای پیمایش پس ترتیبی (postorder) بکار می‌رود.

۳) این پردازه برای پیمایش پیش ترتیبی (preorder) بکار می‌رود.

۴) اگر جای سطرهای ۵ و۶ پردازه را عوض کنیم، پردازه برای پیمایش پیش ترتیبی (preorder) بکار می‌رود.

۵ـ به فرض داشتن یک درخت دودویی کامل با N گره، برای هر گره با اندیس i به گونه‌ای که $1 \le i \le N$ باشد کدام گزینه نادرست است؟

۱) اگر $i \neq 1$ باشد، آنگاه فرزند i در $i/2$ است.

۲) اگر $2i \le N$ باشد، آنگاه فرزند چپ i در $2i$ است.

۳) اگر $2i > N$ باشد، آنگاه i فرزند چپ ندارد.

۴) اگر $2i+1 \le N$ باشد، آنگاه فرزند راست i در $2i+1$ است.

۶ـ یک درخت دودویی در سه آرایه موازی به نام‌های info و left و Right پیاده‌سازی شده به طوری که info به اطلاعات هر گره و Right و left اشاره‌گر به فرزندان راست و چپ آن می‌باشند و Root به ریشه درخت اشاره می‌کند. زیر برنامه بازگشتی زیر چه عملیاتی روی درخت فوق انجام می‌دهد؟

```
A (left , Right , Root ,S)
if  Root = Null  then  S: = 0 & Return
Call A (left , Right , left [Root],S1)
Call A (left , Right , Right [Root],S2)
S: = S1+S2+1
Return
```

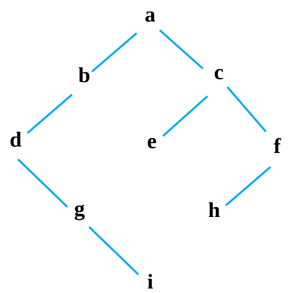
۱) تعداد برگ‌های درخت را پیدا می‌کند.

۲) عمق درخت را پیدا می‌کند.

۳) سطح درخت را پیدا می‌کند.

۴) تعداد گره‌های درخت را پیدا می‌کند.

۷ــ کدام گزینه پیمایش Postorder درخت روبروست؟



۱) dgibehfca

۲) igdbehfca

۳) dgibaechf

۴) abdgicefh

۸ــ تابع زیر چه عملی انجام می‌دهد؟ (t اشاره‌گر به ریشهٔ درخت دودویی است)

```
Treepointer   f(Treepointer   t)
{Treepointer   temp;
if(t!=NULL){
    temp=new Treenode;
    temp→Leftchild= f(t→Leftchild);
    temp→Rightchild= f(t→Rightchild);
    temp→Data= t→Data;
    return temp;
        }
    else return (NULL);
}
```

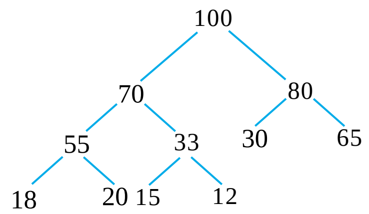
۱) پیمایش پیشوندی (preorder) را انجام می‌دهد.

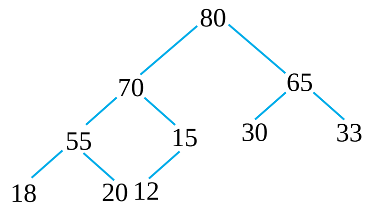
۲) یک کپی از درخت اولیه ایجاد می‌کند.

۳) پیمایش میانوندی (inorder) را انجام می‌دهد.

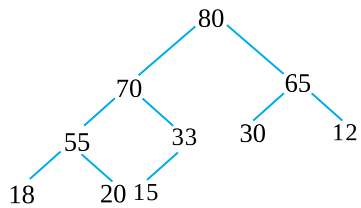
۴) پیمایش پسوندی (postorder) را انجام می‌دهد.

۹ــ اگر یک گره از درخت Max-heap زیر حذف شود، درخت حاصل کدام گزینه است؟

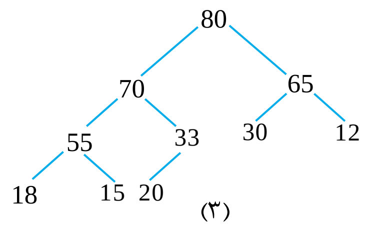




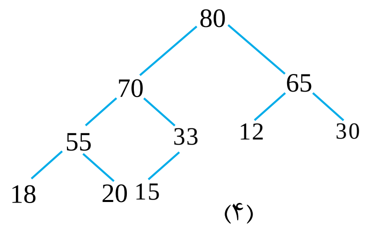
(۱)



(۲)



(۳)

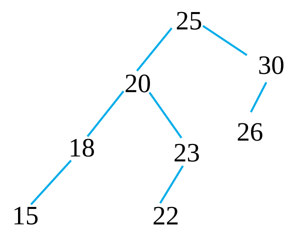


(۴)

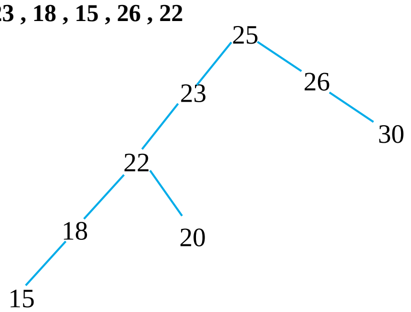
۱۰ــ فرض کنید اعداد زیر به ترتیب در یک درخت جستجوی دودویی خالی درج شوند. درخت نهایی کدام است؟

25 , 30 , 20 , 23 , 18 , 15 , 26 , 22

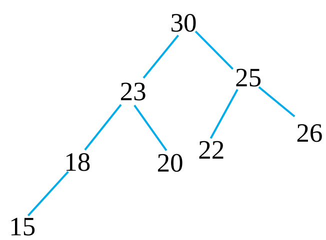
۱)



۲)



۳)

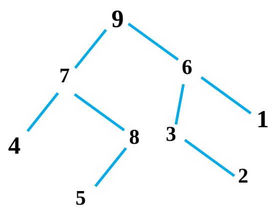


۴) هیچ‌کدام

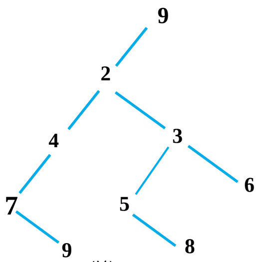
۱۱ـ یک درخت 2- کامل درختی است که هر گره آن صفر یا 2 فرزند دارد. در یک درخت 2- کامل با ارتفاع 7 حداقل و حداکثر گره‌ها کدام گزینه است؟

۱) حداقل 13 گره و حداکثر 127 گره
۲) حداقل 13 گره و حداکثر 64 گره
۳) حداقل 64 گره و حداکثر 127 گره
۴) حداقل 64 گره و حداکثر 128 گره

۱۲ـ دو درخت دودویی 1 و 2 زیر داده شده است:



(۱)



(۲)

کدام‌یک از پیمایش‌ها بر روی درخت 1 و 2 به ترتیب از راست به چپ با یکدیگر یکسان است؟

۱) Postorder , inorder
۲) Postorder , Preorder
۳) Preorder , Preorder
۴) Preorder , inorder

۱۳ـ حداکثر تعداد گره‌های یک درخت دودویی با 10 برگ کدام است؟

۱) 18
۲) 9
۳) نامحدود است.
۴) 21

۱۴ـ عناصر کدام‌یک از آرایه‌های زیر از چپ به راست max-heap نیست؟

۱) $<20, 12, 18, 8, 7, 12, 13, 7, 6, 8, 7, 6>$
۲) $<30, 20, 25, 18, 17, 14, 20, 15, 16, 15, 14, 18>$
۳) $<20, 18, 19, 17, 16, 15, 14, 16, 15, 14, 13, 16>$
۴) $<30, 27, 28, 20, 19, 25, 24, 13, 14, 17, 16, 21>$

۱۵ـ چند max-heap با چهار عنصر که حاوی کلیدهای متمایز یک تا چهار است، می‌توان ساخت؟

۱) 5
۲) 4
۳) 3
۴) 2

۱۶ـ پیمایش preorder یک max-heap به صورت مقابل می‌باشد: **20 , 17 , 15 , 7 , 8 , 14 , 9 , 18 , 13 , 12**

پیمایش inorder آن کدام است؟

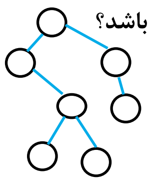
۱) 7 , 15 , 8 , 9 , 17 , 14 , 20 , 13 , 18 , 12
۲) 7 , 15 , 8 , 17 , 9 , 14 , 20 , 13 , 12 , 18
۳) 7 , 14 , 8 , 9 , 17 , 15 , 20 , 13 , 18 , 12
۴) 7 , 15 , 8 , 17 , 9 , 14 , 20 , 13 , 18 , 12

۱۷ـ فرض کنید دو max-heap هر یک دارای k عدد، داده شده باشند، ادغام این دو heap در یک آرایه به صورت نزولی دارای چه مرتبه زمانی است؟

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

۱۸ـ به چند طریق می‌توان اعداد 1 تا 7 را در گره‌های درخت زیر برچسب‌گذاری کرد تا عدد هر گره از اعداد فرزندان آن کوچک‌تر باشد؟

(دقت کنید که از هر 7 عدد باید استفاده شود و تکرار مجاز نیست.)



۱) 28
۲) 30
۳) 32
۴) 34

۱۹ـ یک min-heap حاوی 46 عنصر با کلیدهای 1 تا 46 است. کوچک‌ترین عددی که می‌تواند در آخرین سطح این heap قرار گیرد کدام‌یک از اعداد زیر می‌تواند باشد؟

۱) 5
۲) 7
۳) 6
۴) 32

۲۰ـ کدام‌یک از گزاره‌های زیر در مورد یک درخت 5 تایی کامل با 78 گره صحیح است؟

(با فرض این که ریشه، گره اول باشد و در هر سطح گره‌ها به ترتیب از چپ به راست در نظر گرفته شوند.)

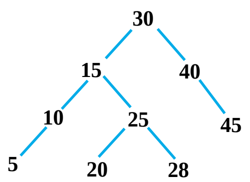
۱) این درخت 62 برگ دارد و گره سیزدهم آن پدر گره شصت و یکم آن است.
۳) این درخت 63 برگ دارد و گره سیزدهم آن پدر گره شصت و یکم آن است.
۲) این درخت 63 برگ دارد و گره سیزدهم آن پدر گره شصت و دوم آن است.
۴) این درخت 62 برگ دارد و گره سیزدهم آن پدر گره شصت و دوم آن است.

فصل ششم

«داده‌ساختارهای پیشرفته»

تست‌های تألیفی فصل ششم

مثال۱: اگر عنصر با کلید 29 به درخت AVL زیر اضافه شود، پیمایش Preorder درخت حاصل کدام گزینه است؟



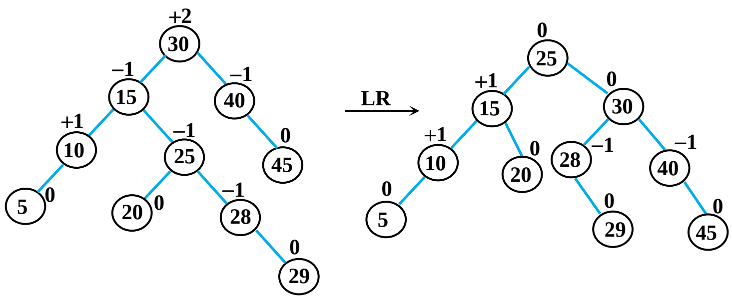
۱) $30,15,10,5,25,20,28,29,40,45$

۲) $30,20,10,5,15,28,25,29,40,45$

۳) $25,15,10,5,20,30,28,29,40,45$

۴) $25,15,10,5,20,30,29,28,40,45$

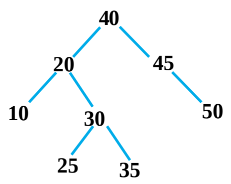
☑ **پاسخ:** گزینه «۳» برای اضافه شدن عنصری با کلید 29 به درخت AVL مورد سؤال، این کلید باید در زیر درخت راست گره‌ی 28 اضافه شود که در این صورت درخت دیگر متوازن نخواهد بود. پس باید با عملیات چرخش، دوباره درخت جستجوی دودویی متوازن شود. بعد از اضافه شدن کلید 29، عامل توازن برای کلید 30، 2+ می‌شود و برای کلید 15، 1- است، پس نیاز به چرخش LR داریم. عملیات مورد نظر در زیر نشان داده شده است:



بعد از عملیات چرخش، درخت به دست آمده یک درخت AVL است و پیمایش Preorder آن به صورت زیر می‌باشد:

$25,15,10,5,20,30,28,29,40,45$

مثال۲: اگر عنصر با کلید 27 به درخت AVL زیر اضافه شود، پیمایش Preorder درخت حاصل کدام گزینه است؟



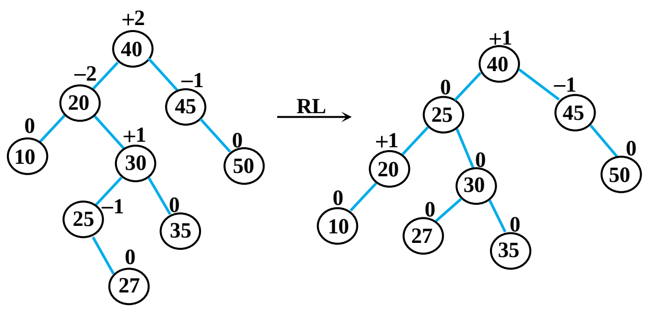
۱) $40,25,20,10,30,27,35,45,50$

۲) $40,20,10,30,25,27,35,45,50$

۳) $30,25,20,10,27,35,40,45,50$

۴) $40,25,20,10,30,27,35,50,45$

☑ **پاسخ:** گزینه «۱» برای اضافه شدن عنصری با کلید 27 به درخت AVL سؤال، این کلید باید در زیر درخت راست گره 25 اضافه شود که در این صورت درخت دیگر متوازن نخواهد بود. پس باید با عملیات چرخش، دوباره درخت جستجوی دودویی متوازن شود، بعد از اضافه شدن کلید 27، عامل توازن برای کلید 20، -2 می‌شود و برای کلید 30، $+1$ است، پس نیاز به چرخش RL داریم. عملیات مورد نظر در زیر نشان داده شده است:



بعد از عملیات چرخش، درخت به دست آمده یک درخت AVL است و پیمایش Preorder آن به صورت زیر می‌باشد:

$40,25,20,10,30,27,35,45,50$

مثال۳: **دو درخت BST دلخواه B_1 و B_2 را در نظر بگیرید، با انجام چند چرخش می‌توان درخت B_1 را به B_2 تبدیل نمود؟**

۱) با انجام $\log_2^n$ چرخش
۲) با انجام $O(n)$ چرخش
۳) با انجام $O(n^2)$ چرخش
۴) در حالت کلی این عمل امکان‌پذیر نیست.

پاسخ: گزینه «۲» هر درخت دودویی را می‌توان با $O(n)$ چرخش به یک درخت اریب (مورب) تبدیل نمود. تبدیل درخت اریب به حالت پیشین نیز به همین تعداد چرخش نیاز دارد. در نتیجه درخت B_1 را می‌توان به یک درخت BST اریب تبدیل کرد و سپس آن درخت BST اریب را به درخت B_2 تبدیل نمود. توجه کنید که درخت اریب متناظر با این دو درخت BST مشابه هستند. تعداد این چرخش‌ها از مرتبه $O(n)$ خواهد بود.

مثال۴: **ساختاری مشابه با درخت AVL درنظر بگیرید که اختلاف ارتفاع دو زیردرخت هر عنصر می‌تواند کوچکترمساوی 2 باشد. در چنین شرایطی حداقل تعداد عناصر این درخت با ارتفاع 8 چه مقدار است؟**

۱) 40
۲) 35
۳) 27
۴) 18

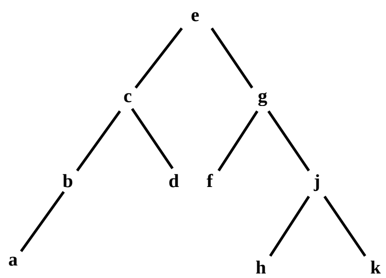
پاسخ: گزینه «۱» اگر a_n را حداقل تعداد عناصر این نوع درخت درنظر بگیریم، رابطه بازگشتی تعداد عناصر این درخت به شکل زیر است:

$$a_n = a_{n-1} + a_{n-3} + 1$$

شرایط اولیه این دنباله به‌صورت $a_0 = 1$ ،$a_1 = 2$ ،$a_2 = 3$ است. مقدار a_8 را می‌توان با تکرار محاسبه نمود که برابر 40 خواهد بود.

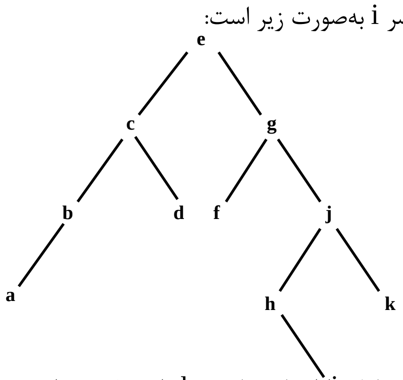
$$a_3 = 5,\ a_4 = 8,\ a_5 = 12,\ a_6 = 18,\ a_7 = 27,\ a_8 = 40$$

مثال۵: **در درخت AVL زیر، فرزند راست عنصر h زیر پس از درج عنصر i چه عنصری خواهد بود؟**

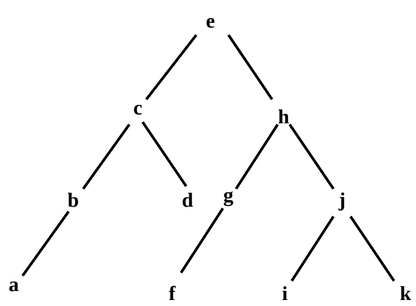


۱) i
۲) j
۳) g
۴) Null

پاسخ: گزینه «۲» شکل اولیه درخت پس از درج عنصر i به‌صورت زیر است:



اختلاف ارتفاع فرزندان g برابر 2 است. در نتیجه می‌بایست تعادل را ایجاد نماییم. h را در فرزند راست e قرار می‌دهیم. فرزند راست h، فرزند راست g و فرزند چپ h عنصر g خواهد بود. گره Null (فرزند چپ h در درخت فوق) در فرزند راست g و گره i (فرزند راست h در درخت فوق) در فرزند چپ j قرار می‌گیرند. در نتیجه به شکل زیر می‌رسیم:



مشاهده می‌شود که فرزند راست گره h، گره j است.

مثال ۶: مرتبه زمانی یافتن k امین کوچکترین عنصر، اضافه کردن یک عنصر جدید و حذف یک داده در درخت AVL با n عنصر کدام است؟

۱) $O(n), O(\log n), O(n)$　　۲) $O(n), O(n), O(\log n)$

۳) $O(\log n), O(\log n), O(\log n)$　　۴) $O(\log n), O(\log n), O(n)$

پاسخ: گزینه «۳» مرتبه زمانی تمامی الگوریتم‌های بیان شده در AVL همگی از مرتبه عمق درخت است.

مثال ۷: عمق مورد انتظار در یک درخت Treap کدام است؟

۱) $\theta(n)$　　۲) $\theta(\log n)$　　۳) $\theta(\log_n^2)$　　۴) $\theta(\frac{n}{2})$

پاسخ: گزینه «۲» عمق مورد انتظار در درخت‌های Treap برابر $\theta(\log n)$ می‌باشد.

مثال ۸: در چه مرتبه زمانی می‌توان تعداد وارونگی‌های موجود در یک آرایه n عنصری را یافت؟

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

پاسخ: گزینه «۳» با استفاده از درخت مرتبه آماری می‌توان در زمان $O(\log_2^n)$ این کار را انجام داد.

مثال ۹: حداکثر تعداد بازه‌های همپوشان از بین n بازه داده شده از اعداد حقیقی را در چه زمانی می‌توان یافت؟

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

پاسخ: گزینه «۲» با استفاده از درخت بازه‌ها می‌توان در زمان $O(\log_2^n)$ محاسبه مورد نظر را انجام داد.

آزمون فصل ششم

۱ـ مرتبه زمانی چرخش‌ها در یک درخت AVL با n گره چقدر است؟

۱) $O(\log_2^n)$　۲) $O(n)$　۳) $O(1)$　۴) $O(n\log_2^n)$

۲ـ مرتبه زمانی یافتن حداکثر تعداد بازه‌های دو به دو ناهمپوشان با داشتن n بازه چقدر است؟

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

۳ـ حداقل تعداد کلیدهای یک $B-tree$ با عمق h کدام است؟ (t نشان‌دهنده مینیمم درجه است)

۱) $1+(t-1)\sum_{i=1}^{h}2t^{i-1}$　۲) $1+\sum_{i=1}^{1}2t^{i-1}$　۳) $\sum_{i=1}^{h}2t^{i-1}$　۴) $(t-1)\sum_{i=1}^{h}2t^{i}$

۴ـ کدام رابطه مرتبه عمق یک درخت $B-tree$ با مینیمم درجه t را به درستی نشان می‌دهد؟

۱) $O(\log_2^n)$　۲) $O(\log_t^n)$　۳) $O(n)$　۴) $O(t)$

۵ـ تعداد گره‌های موجود در عمق i از یک درخت دوجمله‌ای B_k کدام است؟

۱) $\binom{k}{i}$　۲) $\frac{\binom{k}{i}}{2k}$　۳) i^{k-1}　۴) i^k

۶ـ عمل درج در یک درخت heap دوجمله‌ای با n گره در چه زمانی است؟

۱) $O(n)$　۲) $O(\log_2^n)$　۳) $O(1)$　۴) $O(n\log n)$

۷ـ کدام عمل به صورت کارآمد در heapها پشتیبانی نمی‌شود؟

۱) کاهش یک کلید　۲) حذف کوچکترین کلیدها　۳) جستوجو　۴) درج

۸ـ هزینه سرشکن شده عمل اجتماع دو heap ـ فیبوناچی کدام است؟

۱) $\theta(1)$　۲) $(\log_2^n)$　۳) $\theta(n)$　۴) $\theta(n+\log_2^n)$

۹ـ هزینه یافتن اختلاف مقادیر کمینه و بیشینه در یک Deap با n عنصر کدام است؟

۱) $O(1)$　۲) $O(\log n)$　۳) $O(\sqrt{n})$　۴) $O(n)$

۱۰ـ هزینه یافتن یک عنصر در Min-Max Heap با n گره کدام است؟

۱) $O(1)$　۲) $O(\log n)$　۳) $O(\sqrt{n})$　۴) $O(n)$

فصل هفتم

«درهم‌سازی»

آزمون فصل هفتم

۱ـ اگر از روش زنجیره‌سازی برای برطرف نمودن برخورد استفاده شده باشد، آنگاه میانگین زمان جست‌وجوی یک کلید چقدر است؟ (اگر n تعداد کلیدهای نگاشت شده و m اندازه جدول درهم‌سازی باشد).

۱) $\theta(n)$ ۲) $\theta(m)$ ۳) $\theta(\frac{n}{m})$ ۴) $\theta(\frac{m}{n})$

۲ـ در سؤال قبل حداکثر تعداد مقایسه‌ها در جست‌وجو چقدر است؟

۱) n ۲) m ۳) $\frac{n}{m}$ ۴) $\frac{m}{n}$

۳ـ کدام گزینه شکل کلی تابع درهم‌سازی در حالت آدرس‌دهی باز را نشان می‌دهد؟ (فرض کنید U مجموعه کل کلیدهای ممکن و h تابع درهم‌سازی باشد).

۱) $h:U \to U$ ۲) $h:U\times\{0,1,\ldots,m-1\} \to U$

۳) $h:U \to U\times\{0,1,\ldots,m-1\}$ ۴) $h:U \to \{0,1,\ldots,m-1\}$

۴ـ کدام مورد تابع درهم‌سازی مربوط به روش آدرس‌دهی باز با روش وارسی خطی (اگر تابع تقسیم استفاده شده باشد) را نشان می‌دهد؟

۱) $h(k,i)=k \mod m+i$ ۲) $h(k,i)=[(k \mod m)+i]\mod m$

۳) $h(k,i)=k \mod (m+i)$ ۴) $h(k,i)=\mod m+k$

۵ـ اگر از تابع درهم‌سازی مطرح شده در سؤال قبل در یک جدول درهم‌سازی با اندازه 11 استفاده شده باشد، آنگاه در عمل اضافه نمودن چهار کلید زیر چند برخورد رخ خواهد داد؟ (از چپ به راست)
37,83,97,78

۱) دو برخورد ۲) سه برخورد ۳) یک برخورد ۴) برخوردی به وجود نمی‌آید.

۳۹ـ کدام‌یک از جدول‌های درهم‌سازی زیر هنگام درج عناصر 13,8,20,11,7,5 از چپ به راست در جدول تهی بیشترین تعداد تصادم را خواهد داشت؟ (فرض کنید که تصادم به روش درهم‌سازی باز و با وارسی خطی انجام می‌شود.)

۱) اندازهٔ جدول برابر 7 و تابع درهم‌سازی اولیه $H(x)=x \mod 7$.
۲) اندازهٔ جدول برابر 6 و تابع درهم‌سازی اولیه $H(x)=x \mod 6$.
۳) اندازهٔ جدول برابر 8 و تابع درهم‌سازی اولیه $H(x)=x \mod 8$.
۴) اندازهٔ جدول برابر 9 و تابع درهم‌سازی اولیه $H(x)=x \mod 9$.

۷ـ اگر از یک جدول درهم‌سازی با اندازه 11 استفاده شده باشد و تابع درهم‌سازی وارسی خطی با استفاده از تابع تقسیم به کار رفته باشد، آنگاه در صورتی که دنباله کلیدهای زیر از چپ به راست درج شوند، کلید 25 در چه اندیسی قرار می‌گیرد؟
37,83,97,78,14,59,25

۱) 3 ۲) 4 ۳) 5 ۴) 7

۸ـ اگر در روش chaining از لیست‌های پیوندی دوگانه استفاده کنیم، آنگاه زمان درج و حذف یک گره چقدر خواهد بود؟

۱) هر دو $O(1)$ ۲) $O(1),O(\frac{n}{m})$ ۳) $O(\frac{n}{m}),O(1)$ ۴) هر دو $O(\frac{n}{m})$

۹ـ از بین موارد بیان شده در زیر کدام موارد صحیح است؟

الف) فاکتور بارگذاری در روش آدرس‌دهی باز همواره کوچکتر یا مساوی یک است.
ب) در روش chaining در خانه‌های جدول درهم‌سازی کلیدی ذخیره نمی‌شود.

۱) فقط الف ۲) فقط ب ۳) الف و ب ۴) هیچ‌کدام

۱۰ـ اگر اندازه جدول درهم‌سازی برابر m باشد و از روش وارسی خطی استفاده گردد، آنگاه چند دنباله وارسی متفاوت امکان‌پذیر است؟

۱) m ۲) $\frac{n}{m}$ ۳) n ۴) بی‌نهایت

بخش سوم: روش‌های حل مسأله

فصل هشتم

«الگوریتم‌های تقسیم و غلبه (Divide and Conquer)»

تست‌های تألیفی فصل هشتم

مثال ۱: فرض کنید مقادیر دنباله $<y_1,y_2,...,y_n>$ خروجی تابع $f(x_i)$ هستند که f تابع درجه 3 نامشخص و دنباله $<x_1,x_2,...,x_n>$ دنباله‌ای صعودی است. با فرض اینکه بدانیم y_1 عددی مثبت و y_n عددی منفی است، از چه مرتبه زمانی می‌توان ریشه‌ای از تابع درجه 3 را (با استفاده از درون‌یابی) بیابیم؟

۱) $O(\lg n)$ ۲) $O(\lg^2 n)$ ۳) $O(n)$ ۴) $O(\frac{n}{\lg n})$

پاسخ: گزینه «۱» در صورتی که (x_i, y_i) و (x_j, y_j) دو نقطه از مجموعه نقاط روی یک منحنی درجه 3 باشند، اگر مقدار $y_i \times y_j$ یک عدد منفی باشد، حداقل یک x در بازه x_i تا x_j یافت می‌شود که ریشه معادله درجه 3 متناظر با منحنی باشد. شرط منفی بودن مقدار $y_i \times y_j$ در ابتدای مسأله برقرار است. کافی است در هر مرحله، مقداری مانند y_k در وسط این دو مقدار انتخاب کنیم. درصورتی که $y_i \times y_k$ منفی بود، جست‌وجو را در بازه i تا k و در غیر این صورت در بازه k تا j ادامه دهیم. با هر مقایسه، اندازه بازه نصف می‌شود. در نتیجه محاسبه جواب از مرتبه $O(\lg n)$ خواهد بود.

مثال ۲: اگر $S(n)$ و $u(n)$ به ترتیب میانگین تعداد مقایسه‌ها درحالت موفق و ناموفق جست و جوی دودویی باشند، آنگاه از بین موارد زیر کدام نادرست است؟

۱) $S(n) \in \theta(u(n))$ ۲) $S(n) = (1+\frac{1}{n})u(n) - 1$ ۳) $S(n) \in O(\log_2^n)$ ۴) $S(n) \in \theta(n \log n)$

پاسخ: گزینه «۴» صورت درست گزینه چهارم به صورت مقابل است: $S(n) \in \theta(\log_2^n)$

مثال ۳: مرتبه زمانی الگوریتم های ضرب اعداد صحیح بزرگ، ضرب استراسن و یافتن نزدیک‌ترین جفت نقاط در یک صفحه به ترتیب از راست به چپ کدام است؟

۱) $n \log n, n^{\log_2^7}, n^{\log_2^3}$ ۲) $n \log n, n^3, n^{\log_2^3}$ ۳) $n^2, n^{\log_2^3}, n^{\log_2^7}$ ۴) $n^2, n^{\log_2^7}, n \log n$

پاسخ: گزینه «۱» رابطه بازگشتی و مرتبه زمانی مسئله‌های مطرح شده را می‌توان به صورت زیر در نظر گرفت:

ضرب اعداد صحیح بزرگ: $T(n) = 3T(\frac{n}{2}) + \theta(n) \Rightarrow T(n) \in \theta(n^{\log_2^3})$

ضرب استراسن: $T(n) = 7T(\frac{n}{2}) \Rightarrow T(n) \in \theta(n^{\log_2^7})$

یافتن نزدیک‌ترین جفت نقطه: $T(n) = 2T(\frac{n}{2}) + \theta(n) \Rightarrow T(n) \in \theta(n \log n)$

آزمون فصل هشتم

۱ـ از بین موارد زیر کدامیک در مورد جست‌وجوی دودویی برقرار است؟

الف) تعداد مقایسه‌ها در جست‌وجوی موفق همواره $[\log_2^n]$ می‌باشد.	ب) تعداد مقایسه‌ها در جست‌وجوی ناموفق همواره $[\log_2^n]$ می‌باشد.

۱) فقط الف	۲) فقط ب	۳) الف و ب	۴) هیچ‌کدام

۲ـ میانگین تعداد مقایسه‌ها در جست‌وجوی دودویی بر روی یک آرایه 9 عنصری در جست‌وجوهای موفق چقدر است؟

۱) $\frac{25}{9}$	۲) $\frac{31}{9}$	۳) $\frac{30}{9}$	۴) $\frac{24}{9}$

۳ـ مرتبه زمانی الگوریتم ضرب استراسن کدام است؟

۱) $\theta(n^3)$	۲) $\theta(n \log n_3)$	۳) $\theta(n^{\log_2^7})$	۴) $\theta(n \log n_7)$

۴ـ مرتبه زمانی الگوریتم ضرب اعداد صحیح بزرگ n رقمی کدام است؟

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

۵ـ نزدیکترین جفت نقطه از بین n نقطه داده شده را در چه زمانی می‌توان یافت؟

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

۶ـ کدامیک از مسائل زیر با استفاده از ایده تقسیم و غلبه به صورت کارا حل نمی‌شود؟

۱) ضرب ماتریس‌ها	۲) ضرب اعداد صحیح بزرگ

۳) یافتن نزدیکترین جفت نقطه در یک فضای دو بعدی	۴) محاسبه کمترین تعداد ضرب عددی در مسئله ضرب زنجیری ماتریس‌ها

۷ـ کدام‌یک از گزینه‌های زیر رابطه بین میانگین تعداد مقایسه‌ها در جست و جوهای موفق ((s(n)) و ناموفق ((u(n)) در الگوریتم جست و جوی دودویی را به درستی نشان می‌دهد؟

۱) $s(n) = (1+\frac{1}{n})u(n) - 1$	۲) $s(n) = u(n) - 1$	۳) $s(n) = 1 + \frac{1}{n}u(n)$	۴) $s(n) = u(n) + 1$

۸ـ حداقل تعداد مقایسه‌ها در حالت جست‌وجوی ناموفق در جست و جوی دودویی کدام است؟

۱) ۱	۲) $[\log_2^n] - 1$	۳) $[\log_2^n]$	۴) $[\log_2^n] + 1$

۹ـ رابطه بازگشتی مربوط به تعداد جمع و تفریق‌ها در روش استراسن کدام است؟

۱) $T(n) = 7T(\frac{n}{2})$	۲) $T(n) = 18T(\frac{n}{2}) + n$	۳) $T(n) = 7T(\frac{n}{2}) + 18\frac{n}{2}$	۴) $T(n) = 7(\frac{n}{2}) + 18(\frac{n}{2})^2$

۱۰ـ اگر در ضرب دو ماتریس 8×8 از روش استراسن استفاده شود. آنگاه در صورتی که حد آستانه فراخوانی بازگشتی $n = 2$ باشد، تعداد فراخوانی‌های الگوریتم کدام است؟

۱) 56	۲) 57	۳) 49	۴) 48

فصل نهم

«برنامه‌ریزی پویا (Dynamic Programming)»

تست‌های تألیفی فصل نهم

مثال ۱: اگر برای محاسبه $\binom{n}{k}$ از روش تقسیم و غلبه استفاده کنیم، آن‌گاه تعداد فراخوانی‌های الگوریتم کدام است؟

۱) $2\binom{n}{k}$ ۲) $\binom{n}{k}-1$ ۳) $2\binom{n}{k}-1$ ۴) $2\binom{n}{k}-2$

☑ **پاسخ:** گزینه «۳»

مثال ۲: مقدار حافظه مصرفی در الگوریتم برنامه‌ریزی پویا برای محاسبه $\binom{n}{k}$ چقدر است؟

۱) $\theta(nk)$ ۲) $\theta\binom{k}{n}$ ۳) $\theta(k^n)$ ۴) $\theta(n^2)$

☑ **پاسخ:** گزینه «۱» از یک آرایه $n\times k$ (یا $k\times n$) به عنوان جدول پویا استفاده می‌شود.

مثال ۳: کدام رابطه نشان‌دهنده تعداد کل روش‌های ضرب $n+1$ ماتریس به صورت زنجیره‌ای نمی‌باشد؟

۱) $\sum_{k=1}^{n} T(k-1)T(n-k)$ ۲) $\sum_{k=0}^{n-1} T(k)T(n-k-1)$ ۳) $\dfrac{\binom{2n}{n}}{n+1}$ ۴) $n\times\dfrac{\binom{2n-2}{n-1}}{n}$

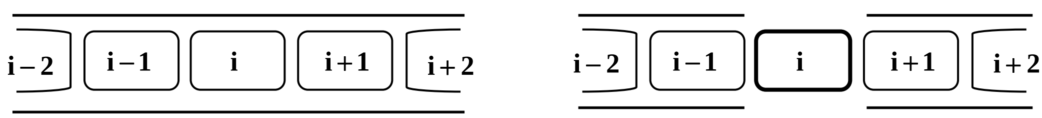
☑ **پاسخ:** گزینه «۴» سه گزینه اول همگی نشان‌دهنده عدد کاتالان C_n می‌باشند.

مثال ۴: اگر مرتبه زمانی محاسبه درخت BST بهینه با داشتن n کلید و یافتن ترتیب بهینه ضرب در ضرب n ماتریس به صورت زنجیره‌ای با استفاده از روش برنامه‌ریزی پویا را به ترتیب با $B(n)$ و $M(n)$ نشان دهیم، آن‌گاه کدام مورد صحیح است؟

۱) $B(n)\in\theta(n^2)$ ۲) $M(n)\in\theta(n^2\log n)$ ۳) $B(n)\in\Omega(M(n))$ ۴) $B(n)\in\theta(M(n))$

☑ **پاسخ:** گزینه «۴» هر دو الگوریتم از مرتبه $\theta(n^3)$ می‌باشند.

مثال ۵: ماده‌ای فاسد شدنی در اختیار داریم که از کنارهم قرار گرفتن n تکه در یک ردیف تشکیل شده است. نوعی دستگاه نیز در اختیار داریم که این ماده را به‌گونه‌ای برش می‌دهد که تکه iام را بسته‌بندی و ماده باقی‌مانده را به دو تکه از محل قرارگرفتن ماده iام تقسیم می‌کند که این کار یک ساعت طول خواهد کشید. درصورتی که ارزش هر تکه از ماده مشخص و متفاوت با بقیه تکه‌ها باشد، محدودیتی در تعداد دستگاه‌ها وجود نداشته باشد و ارزش تکه‌های فاسد شدنی در هر ساعت، به‌اندازه یک‌سوم کاهش یابد، از چه مرتبه‌ای می‌توان ترتیب محل‌های برش را مشخص نمود بطوری که مجموع ارزش تکه‌های بسته‌بندی شده بیشینه شود؟ شکل زیر نحوه عملکرد دستگاه در تقسیم نمودن ماده سمت چپ به دو تکه و بسته‌بندی شدن تکه iام را نشان می‌دهد.



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

☑ **پاسخ:** گزینه «۴» این مسأله را می‌توان مشابه با مسأله ساخت درخت جست‌وجوی دودویی درنظر گرفت. تکه‌ای که در ابتدا بسته‌بندی می‌شود نقش ریشه را دارد و با توجه به زمان سپری شده، دو قطعه جدا شده مانند دو زیردرخت برای ریشه خواهند بود. این مسأله از روش برنامه‌نویسی پویا از مرتبه $O(n^3)$ قابل حل است.

آزمون فصل نهم

۱ـ مرتبه زمانی یافتن ترتیب بهینه برای ضرب زنجیری ماتریس‌ها کدام است؟

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

۲ـ کدام‌یک از موارد زیر صحیح است؟

الف) روش برنامه‌نویسی پویا یک روش پایین به بالا می‌باشد.

ب) در روش برنامه‌نویسی پویا با "memoization" ابتدا مسائل با اندازه بزرگ‌تر حل می‌شوند.

۱) فقط الف　۲) فقط ب　۳) الف و ب　۴) هیچ‌کدام

۳ـ مرتبه زمانی روش برنامه‌نویسی پویا برای محاسبه جمله nام از سری فیبوناچی از چه مرتبه‌ای است؟

۱) $\theta(2^n)$　۲) $\theta(2^{\frac{n}{2}})$　۳) $\theta(n)$　۴) $\theta(n^2)$

۴ـ مقدار فضای مصرفی کمکی در روش برنامه‌نویسی پویا برای یافتن BST بهینه چقدر است؟

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

۵ـ از بین موارد زیر کدام‌یک در مورد روش برنامه‌نویسی پویا صحیح است؟

الف) یک روش پایین به بالا است.

ب) رئوس گراف زیر مسائل براساس پیمایش dfs ملاقات می‌شوند.

۱) فقط الف　۲) فقط ب　۳) الف و ب　۴) هیچ‌کدام

۶ـ از بین موارد زیر کدام‌یک صحیح می‌باشد؟

الف) مسئله کوله‌پشتی صفر و یک با استفاده از روش برنامه‌نویسی پویا در زمان چند جمله‌ای قابل حل می‌باشد.

ب) مسئله خردکردن پول با استفاده از روش برنامه‌نویسی پویا در زمان چند جمله‌ای قابل حل می‌باشد.

۱) فقط ب　۲) فقط الف　۳) الف و ب　۴) هیچ‌کدام

۷ـ یافتن بزرگ‌ترین زیر دنباله مشترک برای دو رشته X و Y که هر کدام دارای طول n می‌باشند دارای چه مرتبه زمانی است؟

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

۸ـ از بین موارد مطرح شده در زیر کدام مورد (موارد) صحیح است؟

الف) روش برنامه‌ریزی پویا یک روش بالا به پایین است.

ب) روش تقسیم و غلبه یک روش پایین به بالا است.

ج) الگوریتم‌های روش برنامه‌ریزی پویا به صورت بازگشتی می‌باشند.

۱) فقط الف　۲) ب و ج　۳) فقط ج　۴) هیچ‌کدام

۹ـ کدام مورد در رابطه با برنامه‌ریزی پویا با memoization صحیح است؟

۱) یک روش پایین به بالا است.　۲) ممکن است برخی از زیر مسائل را چند بار حل کند.

۳) رئوس زیرگراف مسائل به صورت dfs پیمایش می‌شود.　۴) همواره منجر به زمان خطی می‌گردد.

۱۰ـ مرتبه زمانی کدام‌یک از الگوریتم‌های زیر به روش برنامه ریزی پویا با سایر الگوریتم‌ها متفاوت است؟

الف ـ ضرب زنجیری ماتریس‌ها　ب ـ الگوریتم فلوید ـ وارشال

ج ـ محاسبه BST بهینه　د ـ فروشنده دوره‌گرد

۱) ضرب زنجیری ماتریس‌ها　۲) محاسبه BST بهینه　۳) فروشنده دوره‌گرد　۴) الگوریتم فلوید وارشال

فصل دهم

«الگوریتم‌های حریصانه (Greedy Algorithms)»

تست‌های تألیفی فصل دهم

مثال۱: در مسأله زمان‌بندی کارها، n پردازنده در اختیار داریم و باید m کار را پردازش کنیم (n<<m). با فرض اینکه پردازنده‌ها مشابه باشند و زمان مورد نیاز برای پردازش کار iام برابر c_i باشد (کارها با توجه به مقدار زمان اجرایشان به ترتیب از بیشترین زمان اجرا تا کمترین زمان اجرا مرتب شده اند) و نحوه تخصیص کارها بصورت حریصانه انجام گیرد بطوری که کاری که زمان بیشتری نیاز دارد، در ابتدا پردازش شود، کدام گزینه کران بالای پایین‌تری برای بیشینه زمان پردازش این کارها می‌باشد؟

۱) $\dfrac{\sum_{i=1}^{m-1} c_i}{n} + c_m$

۲) $\dfrac{\sum_{i=1}^{m-n} c_i}{n} + c_{m-n+1}$

۳) $\max\limits_{1\le k\le n} \{\dfrac{\sum_{i=1}^{m-k} c_i}{n} + c_{m-k+1}\}$

۴) $\dfrac{\sum_{i=2}^{m} c_i}{n} + c_1$

پاسخ: گزینه «۴» در چنین شرایطی نمی‌توان بدترین حالت را زمانی دانست که دقیقاً در لحظه شروع پردازش p کار آخر این مجموعه کارها $(p \le n)$ کار همه پردازنده‌ها به اتمام رسیده باشد؛ زیرا ممکن است اختلاف زمان کارهای پیشین با کارهای آخر به حدی باشد که تاثیر زمانی کارهای آخر از بین برود. به‌عنوان مثال درنظر بگیرید از مجموعه m کار، m-n کار به زمان اجرای k واحد زمانی و n کار به زمان اجرای 1 واحد زمانی نیاز داشته باشند و در لحظه شروع کار شماره m-n، پردازش همه پردازنده‌ها باهم به اتمام رسیده باشد. در این حالت برای پردازش کار شماره m-n با k واحد زمانی و نیاز داریم درحالی که n کار باقی‌مانده در 2 واحد زمانی بعد از زمان شروع کار شماره m-n به پایان می‌رسند و پردازش کار شماره m-n تا k-2 واحد زمانی بعد ادامه دارد. در حالت کلی با توجه به اینکه ممکن است اختلاف زمان پردازش کارها زیاد باشد، احتمال اینکه یکی از کارهای ابتدایی یا میانی، آخرین کار به پایان رسیده باشد نیز وجود دارد. در واقع هیچ‌یک از مقادیر گزینه‌های (۱) تا (۳)، نمی‌توانند کران بالایی برای این زمان اجرا باشند. تنها می‌توان مقدار زیر را به‌عنوان حد بالای این زمان درنظر گرفت:

$$\frac{\sum_{i=2}^{m} c_i}{n} + c_1$$

به‌عنوان یک مثال درنظر بگیرید 3 پردازنده در اختیار داریم و 101 کار را می‌بایست انجام دهیم. زمان مورد نیاز برای کار اول برابر 1000 واحد زمانی و برای 100 کار دیگر برابر 1 واحد زمانی است. پردازنده اول برای کار اول به 1000 واحد زمان نیاز دارد و دو پردازنده دیگر، سایر کارها را در 50 واحد زمانی به اتمام می‌رسانند. در نتیجه زمان پردازش کل کارها برابر با زمان پردازش کار اول است.

مثال۲: در مسأله زمان‌بندی کارها، n پردازنده در اختیار داریم و باید m کار را پردازش کنیم (n<<m). با فرض اینکه همه کارها مشابه بوده و زمان پردازش یک کار برای پردازنده i برابر i واحد زمان باشد و تخصیص کار به‌صورت حریصانه با اولویت سریع‌ترین پردازنده باشد (پردازنده‌ای که زمان کمتری نیاز دارد اولویت دارد)، بیشینه زمان پردازش این کارها چه مقدار خواهد بود؟

۱) $\dfrac{m+n-1}{\sum_{i=1}^{n}\frac{1}{i}}$

۲) $\dfrac{m+\frac{1}{n}}{\sum_{i=1}^{n}\frac{1}{i}}$

۳) $\dfrac{m-1+\frac{1}{n}}{\sum_{i=1}^{n}\frac{1}{i}}+n-1$

۴) $\dfrac{m-1}{\sum_{i=1}^{n}\frac{1}{i}}+n$

پاسخ: گزینه «۳» بدترین حالت زمانی مربوط است به هنگامی‌که تنها یک کار باقی بماند، پردازش پردازنده آخر به اتمام برسد و از زمان پردازش تمام پردازنده‌های دیگر، یک واحد زمانی باقی مانده باشد. (در حالتی که m عددی اول باشد، این حالت ممکن است اتفاق بیافتد.) در این حالت، آخرین کار به پردازنده آخر داده خواهد شد. زمان سپری‌شده تا این لحظه برابر t_1 و زمان باقی‌مانده برابر $t_2 = n$ می‌باشد. در نتیجه بیشینه زمان پردازش این کارها برابر است با:

$$t_1 + t_2 = \frac{(m-1) - \sum_{i=1}^{n-1}\frac{1}{i}}{\sum_{i=1}^{n}\frac{1}{i}} + n = \frac{m-1+\frac{1}{n}}{\sum_{i=1}^{n}\frac{1}{i}} + n - 1$$

برای درک بهتر از رابطه فوق، درنظر بگیرید در یک واحد زمانی بعد از لحظه t_1 تعداد m-1 کار بطور کامل به اتمام رسیده است و $\frac{1}{n}$ از زمان اجرای کار mام گذشته است. تمام پردازنده‌ها تا این لحظه مشغول به‌کار بوده‌اند و با توجه به اینکه پردازنده iام در هر لحظه $\frac{1}{i}$ از پردازش یک کار را انجام می‌داد، در هر لحظه، $\sum_{i=1}^{n}\frac{1}{i}$ از مجموعه کارها پردازش می‌شد. پس از آن نیز $\frac{n-1}{n}$ از پردازش پردازنده nام باقی می‌ماند که n-1 واحد زمانی طول می‌کشد. عبارت گزینه (۴) مربوط به حالتی است که اولویت با کندترین پردازنده می‌باشد.

مثال۳: اگر کدگذاری‌های داده شده در مثال قبل را داشته باشیم، آنگاه با در نظر گرفتن فراوانی‌های زیر برای کاراکترها، نسبت میانگین تعداد بیت‌های کد در کدگذاری «ج»، نسبت به کدگذاری «ب» چقدر است؟

کاراکتر	e	d	c	b	a
فراوانی	0.05	0.18	0.20	0.25	0.32

۱) $\frac{2.25}{2.23}$ ۲) $\frac{2.23}{2.25}$ ۳) $\frac{2.8}{2.42}$ ۴) $\frac{2.42}{2.8}$

☑ **پاسخ:** گزینه «۱» اگر از کدگذاری «ب» استفاده شود میانگین تعداد بیت عبارتست از:

$$.32\times 2+.25\times 2+.20\times 3+.18\times 2+.50\times 3=2.25$$

و اگر از کدگذاری «ج» استفاده شود برابر خواهد بود با:

$$.32\times 2+.25\times 2+.20\times 2+.18\times 3+.50\times 3=2.23$$

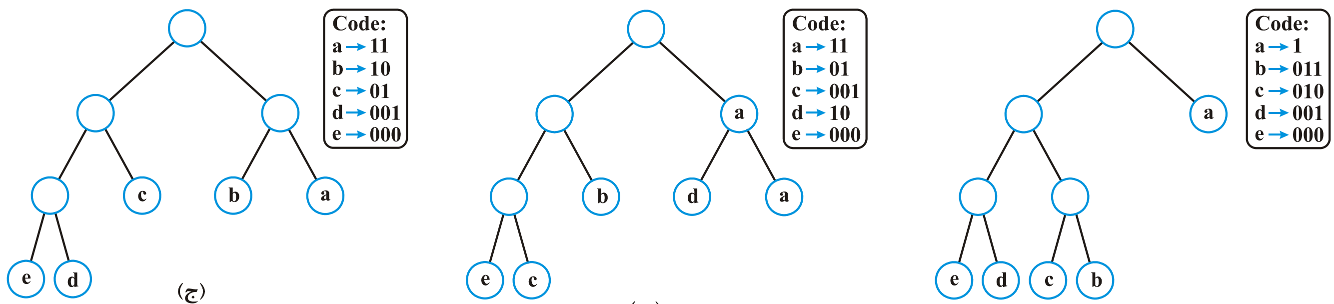
بنابراین نسبت مورد نظر عبارتست از:

$$\frac{2.25}{2.23}$$

مثال۴: تعداد گره‌های سطح سوم در درخت‌های متناظر با کدهای «الف»، «ب» و «ج» به ترتیب چقدر است؟

۱) 2، 2، 3 ۲) 2، 4، 4 ۳) 2، 3، 3 ۴) 2، 2، 4

☑ **پاسخ:** گزینه «۲» درخت‌های مربوط به سه کدگذاری موردنظر به صورت زیر خواهد بود:



(الف) (ب) (ج)

آزمون فصل دهم

۱ـ از بین مسائل زیر چند مورد دارای راه‌حل حریصانه می‌باشند؟

الف) مسئله خرد کردن پول ب) یافتن درخت پوشای کمینه ج) کوله‌پشتی صفر و یک د) کوله‌پشتی کسری

۱) 1 ۲) 2 ۳) 3 ۴) 4

۲ـ مرتبه زمانی الگوریتم حریصانه برای مسئله زمان‌بندی با مهلت معین کدام است؟

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

۳ـ تعداد گره‌های تک‌فرزندی و تعداد گره‌های برگ در درخت کدگذاری هافمن برای n کاراکتر، به ترتیب از راست به چپ کدام است؟

۱) صفر و n ۲) صفر و n-1 ۳) n و n-1 ۴) n-1 و 1

۴ـ اگر برای پیاده‌سازی صف اولویت از یک آرایه نامرتب استفاده شود، آنگاه مرتبه زمانی الگوریتم کدگذاری هافمن کدام است؟

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

۵ـ درخت متناظر با کدگذاری (پیشوندی) بهینه مربوط به یک الفبا را در نظر بگیرید. در این صورت حداکثر و حداقل تعداد گره‌های تک‌فرزندی آن درخت کدام است؟

۱) حداکثر 2 و حداقل صفر ۲) حداکثر 2 و حداقل یک ۳) حداکثر 1 و حداقل صفر ۴) همواره صفر

فصل یازدهم

«الگوریتم‌های مبتنی بر جست‌وجوی درخت فضای حالت»

آزمون فصل یازدهم

۱ـ کدام شرط نشان‌دهنده این است که دو وزیر یکدیگر را تهدید می‌کنند (r_1 و r_2 به ترتیب شماره سطرهای دو وزیر و c_1 و c_2 شماره ستون‌های دو وزیر را نشان می‌دهد).

۱) $c_1 - c_2 = r_1 - r_2$　　۲) $c_2 - c_1 = r_1 - r_2$　　۳) $|c_1 - c_2| = |r_1 - r_2|$　　۴) $|c_1 - c_2| = |r_1 - r_2 + 1|$

۲ـ اگر کل درخت فضای حالت را در مسئله n−وزیر در نظر بگیریم، آنگاه تعداد گره‌های آن چقدر می‌باشد؟

۱) $\frac{n^{n+1}-1}{n-1}$　　۲) $n!$　　۳) n^{n+1}　　۴) $n^n - 1$

۳ـ مرتبه زمانی الگوریتم عقب‌گرد برای حل مسئله کوله‌پشتی صفر و یک کدام است؟

۱) $O(n)$　　۲) $O(n^2)$　　۳) $O(2^n)$　　۴) $O(nW)$

۴ـ تعداد کل گره‌های درخت فضای حالت در روش عقب‌گرد برای حل مسئله حاصل جمع زیر مجموعه‌ها کدام است؟ (تعداد قطعات را n در نظر بگیرید)

۱) 2^n　　۲) $2^n - 1$　　۳) 2^{n+1}　　۴) $2^{n+1} - 1$

۵ـ از بین موارد بیان شده در زیر کدام موارد صحیح است؟

الف) در روش شاخه و قید جست‌وجوی درخت فضای حالت به صورت **preorder** است.

ب) روش شاخه و قید در مورد مسائل بهینه‌سازی استفاده می‌گردد.

۱) فقط الف　　۲) فقط ب　　۳) الف و ب　　۴) هیچ‌کدام

۶ـ مرتبه زمانی الگوریتم عقب‌گرد برای مسئله حاصل‌جمع زیر مجموعه‌ها کدام است؟

۱) $O(n)$　　۲) $O(n^2)$　　۳) $O(2^n)$　　۴) $O(n^n)$

۷ـ در درخت فضای حالت، ریشه نشان‌دهنده کدام مورد است؟

۱) نشان‌دهنده کل مسئله است که در آن هیچ انتخابی صورت نگرفته است.

۲) نشان‌دهنده یک حل کاندید است.

۳) نشان‌دهنده حل بهینه است.

۴) نشان‌دهنده زیر مسئله‌ای است که در آن یک انتخاب انجام شده است.

۸ـ اگر در مسئله حاصل‌جمع زیر مجموعه‌ها داشته باشیم $S = \{5,10,12,13,15,18\}$ و $W = 30$ ، آنگاه چند حل وجود دارد؟

۱) 2　　۲) 3　　۳) 4　　۴) 1

۹ـ در چند مورد از مسائل زیر، حل‌ها در گره‌های موجود در پایین‌ترین سطح درخت فضای حالت قرار دارند؟

الف) حاصل‌جمع زیرمجموعه‌ها　　ب) دور هامیلتونی　　ج) n ـ وزیر

۱) 2　　۲) 3　　۳) 1　　۴) 0

۱۰ـ اگر در مسئله کوله‌پشتی صفر و یک، حل بهینه برداشتن تمام قطعات باشد، آنگاه مرتبه زمانی روش عقب‌گرد کدام است؟

۱) $\theta(2^n)$　　۲) $\theta(2^{n-1})$　　۳) $\theta(n)$　　۴) $\theta(n^2)$

بخش چهارم: الگوریتم‌های گراف

فصل دوازدهم

«الگوریتم‌های پیمایش گراف»

تست‌های تألیفی فصل دوازدهم

مثال۱: اگر یال (u,v) در گراف وجود داشته باشد، آن گاه کدام مورد در رابطه با فاصله u و v از رأس منبع صحیح است؟

۱) $v.d \le u.d+1$ ۲) $v.d = u.d+1$ ۳) $v.d \ge u.d$ ۴) $v.d \le u.d-1$

☑ **پاسخ:** گزینه «۱» با توجه به وجود یال از u به v، فاصله v از منبع حداکثر یک واحد بیشتر از فاصله u از منبع است. با توجه به اینکه ممکن است به واسطه مسیر دیگری از منبع به v برسیم، ممکن است فاصله v.d از u.d کمتر هم باشد.

مثال۲: اگر گراف بوسیله ماتریس همسایگی پیاده‌سازی شده باشد، آنگاه مرتبه زمانی الگوریتم BFS بر روی یک گراف n رأسی کدام است؟

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

☑ **پاسخ:** گزینه «۲» هزینه پیمایش BFS در حالتی که از ماتریس همسایگی استفاده شود $O(n^2)$ خواهد بود.

مثال۳: گراف جهت‌داری با V رأس و E یال در اختیار داریم که وزن همه یال‌ها، برابر عدد مثبت k است. می‌خواهیم طول کوتاه‌ترین مسیر از راس v به سایر رئوس را محاسبه نماییم. از چه مرتبه‌ای می‌توان این کار را انجام داد؟

۱) $O(V)$ ۲) $O(V+E)$ ۳) $O(E+V\log V)$ ۴) $O(E\log V)$

☑ **پاسخ:** گزینه «۲» با توجه به اینکه وزن تمام یال‌ها برابر است، می‌توان از الگوریتم اول سطح برای محاسبه فاصله راس v از سایر رئوس استفاده نمود. پیچیدگی محاسباتی این الگوریتم از مرتبه $O(V+E)$ است.

مثال۴: یک DAG با n رأس، حداکثر چند یال پیش‌سو (forward edge) دارد؟

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

☑ **پاسخ:** گزینه «۴» فرض کنید $\{v_1,v_2,\dots,v_n\}$ رئوس این DAG باشند. در صورتی که این گراف حداکثر تعداد ممکن رئوس را داشته باشد و یال‌های $\{\{v_1,v_2\},\{v_2,v_3\},\dots,\{v_{n-1},v_n\}\}$ یال‌های درختی این گراف درنظر گرفته شود، مابقی یال‌های این گراف، یال پیش‌سو خواهند بود. حداکثر تعداد یال‌های یک DAG با n رأس برابر $\frac{n(n-1)}{2}$ است. $n-1$ یال از این یال‌ها، درختی هستند. در نتیجه حداکثر تعداد یال‌های پیش‌سوی یک DAG با n رأس برابر است با:

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

مثال۵: گرافی بدون جهت با v رأس و e یال را درنظر بگیرید که حداکثر درجه هر رأس آن برابر d است. در این صورت کدام عبارت صحیح خواهد بود؟

۱) در پیمایش bfs این گراف حداکثر تعداد رئوسی که در صف قرار می‌گیرند برابر 3d است.
۲) در پیمایش dfs این گراف حداکثر تعداد رئوسی که در پشته قرار می‌گیرند برابر 2d است.
۳) بررسی وجود دور در این گراف همواره از مرتبه $O(v)$ است.
۴) بررسی همبند بودن این گراف همواره از مرتبه $O(v)$ است.

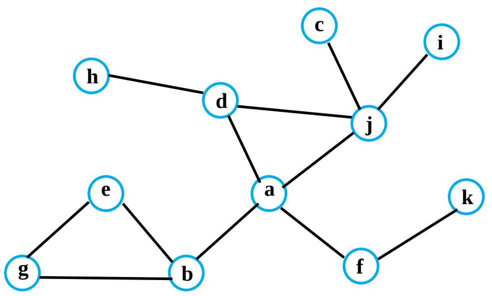
☑ **پاسخ:** گزینه «۳» به ازای $d>1$ حداکثر تعداد عناصری که در پیمایش bfs وارد صف می‌شوند از مرتبه $O(n)$ است. تعداد عناصری که وارد پشته می‌شوند نیز ارتباطی با d ندارد و از مرتبه $O(n)$ خواهد بود. بررسی وجود دور در گراف بدون جهت با استفاده از پیمایش dfs و پس از بررسی دقیقاً v یال ممکن است. ولی بررسی همبند بودن گراف نیاز به پیمایش bfs یا dfs گراف از مرتبه $O(v+e)$ دارد.

مثال۶: درخت با یال‌های بدون جهت و وزن‌دار را درنظر بگیرید. یکی از رئوس این درخت (با برچسب r) را به‌عنوان ریشه درنظر می‌گیریم و درخت را به یک درخت ریشه‌دار تبدیل می‌کنیم. به طوری که اگر u و v دو رأس مجاور در این درخت باشند و تعداد رئوس بین u و r کمتر از تعداد رئوس بین v و r باشد، جهت یال بین این دو رأس، از u به v خواهد بود. (u والد v خواهد بود.) از چه مرتبه‌ای می‌توان برگی مانند t در این درخت را یافت که مسیر از r تا t دارای بیشترین وزن ممکن باشد؟

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

☑ **پاسخ:** گزینه «۴» درخت ریشه‌دار n رأسی با یک مرتبه اجرای الگوریتم BFS روی درخت اولیه قابل ساخت است. هرچند در ادامه حل مسأله نیازی به درخت ریشه‌دار نداریم. کافی است یک مرتبه الگوریتم DFS را روی درخت با شروع از رأس r اجرا کنیم و با رسیدن به هر گره، مجموع هزینه رسیدن تا آن گره را محاسبه نماییم. این هزینه برابر است با مجموع هزینه رسیدن تا والد آن گره به‌علاوه هزینه یال بین گره والد تا گره مذکور. الگوریتم DFS بر روی درخت، مرتبه اجرایی $\theta(n)$ دارد.

مثال۷: گراف زیر را با شروع از رأس a پیمایش می‌کنیم. در این صورت زمان ملاقات کدام رأس در پیمایش DFS زودتر از زمان ملاقات آن رأس در پیمایش BFS است؟ (ترتیب ملاقات براساس حروف الفبا است.)



۱) k
۲) j
۳) c
۴) e

☑ **پاسخ:** گزینه «۳» پیمایش DFS گراف بصورت abegdhjecfk و پیمایش BFS آن به‌صورت abdfjeghkci است. گره c در پیمایش DFS نهمین گره پیمایش شده و در پیمایش BFS دهمین گره پیمایش شده است.

مثال۸: مرتبه زمانی الگوریتم‌های BFS و DFS به ترتیب کدام است؟

۱) هر دو $O(V+E)$	۲) هر دو $\theta(V+E)$	۳) $O(V+E)$ و $\theta(V+E)$	۴) $\theta(V+E)$ و $O(V+E)$

☑ **پاسخ:** گزینه «۳» در الگوریتم BFS لزوماً تمام رئوس و یال‌ها بررسی نمی‌شوند، اما در الگوریتم DFS تمام یال‌ها و رئوس بررسی می‌شوند.

آزمون فصل دوازدهم

۱ـ الگوریتم مرتب‌سازی توپولوژیک از کدام‌یک از الگوریتم‌های زیر استفاده می‌کند؟

۱) BFS　۲) DFS　۳) Floyd　۴) Prim

۲ـ مرتبه زمانی یافتن مؤلفه‌های همبند قوی در یک گراف کدام است؟

۱) $\theta(|V|+|E|)$　۲) $\theta(|E|^2)$　۳) $\theta(|V|^2)$　۴) $\theta(|E|\log|E|)$

۳ـ اگر V_i و V_j دو رأس در یک گراف باشند، به طوری که رأس V_i زودتر از V_j در صف قرار گرفته باشد، در این صورت کدام مورد در رابطه با فاصله این رأس از منبع برقرار است؟

۱) $V_i.d \le V_j.d$　۲) $V_i.d \ge V_j.d$　۳) $V_i.d \ne V_j.d$　۴) $V_i.d = V_j.d$

۴ـ در الگوریتم BFS کدام نوع یال هیچ‌گاه ایجاد نمی‌شود؟

۱) back　۲) forward　۳) tree　۴) cross

۵ـ اگر یک گراف جهت‌دار دور نداشته باشد، آنگاه کدام نوع یال در پیمایش DFS آن دیده نخواهد شد؟

۱) tree　۲) cross　۳) forward　۴) back

۶ـ زمان اجرای مرتب‌سازی توپولوژیک کدام است؟

۱) $\theta(|V|)$　۲) $\theta(|E|)$　۳) $\theta(|V|+|E|)$　۴) $\theta(|V|\log|E|)$

۷ـ برای یافتن مؤلفه‌های همبند قوی در یک گراف، چند بار الگوریتم DFS فراخوانی می‌شود؟

۱) 4 بار　۲) 3 بار　۳) 2 بار　۴) 1 بار

۸ـ اگر یال (u,v) در گراف وجود داشته باشد، آنگاه کدام مورد در رابطه با فاصله u و v از رأس آغازین برقرار است؟

۱) $v.d \ge u.d$　۲) $v.d \le u.d+1$　۳) $v.d = u.d+1$　۴) $v.d \le u.d-1$

۹ـ مرتبه زمانی الگوریتم DFS در صورتی که گراف با استفاده از ماتریس پیاده‌سازی شده باشد، کدام است؟

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

۱۰ـ ساختمان داده مورد استفاده در DFS و BFS به ترتیب عبارتند از:

۱) پشته ـ صف اولویت　۲) پشته ـ صف FIFO　۳) صف FIFO ـ صف اولویت　۴) پشته ـ پشته

۱۱ـ کدام گزینه در مورد مرتب‌سازی توپولوژیک برقرار نیست؟

۱) یک گراف ممکن است مرتب‌سازی توپولوژیک نداشته باشد.

۲) مرتب‌سازی توپولوژیک بر مبنای پیمایش DFS به دست می‌آید.

۳) هر DAG دقیقاً یک مرتب‌سازی توپولوژیک دارد.

۴) اگر گراف دارای دور باشد، آنگاه نمی‌تواند مرتب‌سازی توپولوژیک داشته باشد.

۱۲ـ کدام‌یک از انواع یال‌ها در پیمایش BFS به وجود نمی‌آید؟

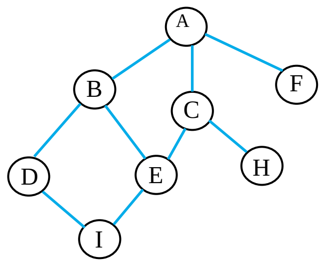
۱) forward　۲) back

۳) tree　۴) هر سه می‌تواند وجود داشته باشد.

۱۳ـ مرتبه اجرای BFS در یک گراف اسپارس کدام است؟

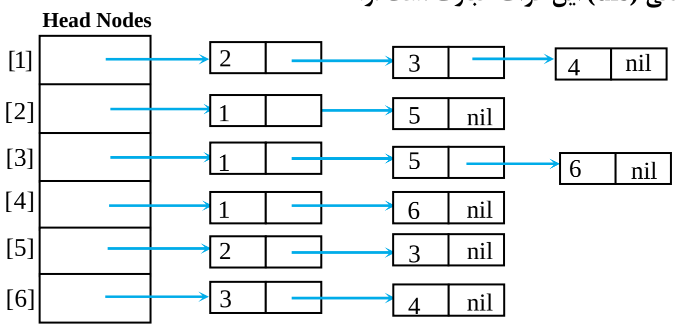
۱) $O(V)$　۲) $O(V^2)$　۳) $O(VE)$　۴) $O(V\log V)$

۱۴ـ اگر در گراف زیر جستجو در عمق (dfs) را از رأس C شروع کنیم، کدام گره‌ها به ترتیب از چپ به راست رؤیت می‌شوند؟
(فرض کنید فرزندان یک گره براساس ترتیب حروف الفبا انتخاب شوند.)



۱) ABCDEFHI
۲) CABDIEFH
۳) CAEHBFID
۴) CABDEHIF

۱۵ـ لیست مجاورتی گراف بدون جهت زیر را در نظر بگیرید، جستجوی عمقی (dfs) این گراف عبارت است از:

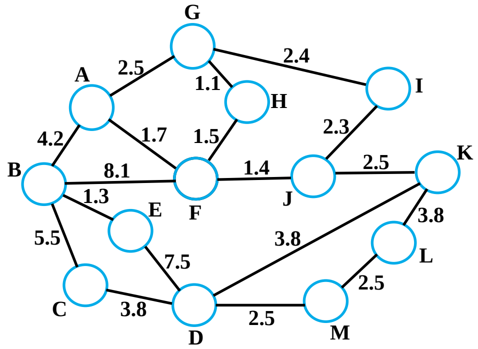


۱) 1,2,5,3,4,6
۲) 1,3,5,6,4,2
۳) 1,2,5,3,6,4
۴) 1,2,5,6,3,4

فصل سیزدهم
«الگوریتم‌های مبتنی بر فاصله در گراف»

تست‌های تألیفی فصل سیزدهم

مثال۱: گراف وزن‌دار زیر مفروض است. کمترین وزن یک درخت پوشا(**minimum weight spanning tree**) برای این گراف برابر است با:



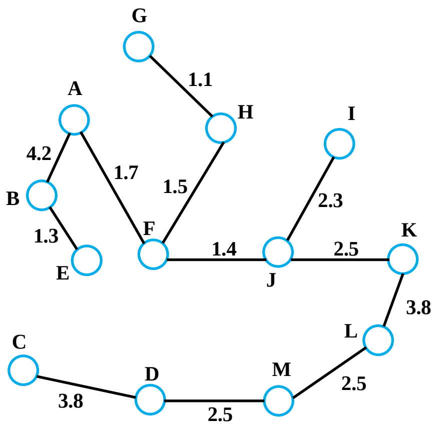
۱) 28.6

۲) 26.9

۳) 30.2

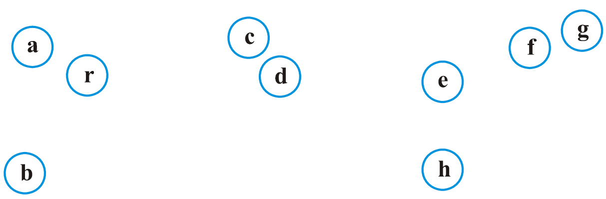
۴) 29.9

☑ **پاسخ:** گزینه «۱» از روش پریم برای پیدا کردن MST استفاده می‌کنیم بنابراین درخت زیر را خواهیم داشت.



که وزن آن برابر 28.6 است.

مثال۲: اگر رأس‌های زیر نشان‌دهنده رأس‌های یک گراف کامل باشند که وزن یال بین هر دو رأس برابر با فاصله اقلیدسی آن دو رأس باشد، آنگاه چهارمین یالی که در الگوریتم کروسکال انتخاب می‌شود کدام است؟



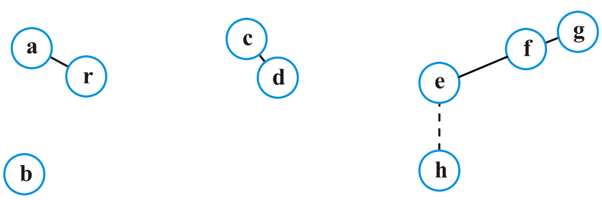
۱) (a,r)

۲) (f,g)

۳) (e,h)

۴) (d,e)

☑ **پاسخ:** گزینه «۳» نحوه انتخاب یال‌ها در الگوریتم کروسکال به صورت زیر خواهد بود.

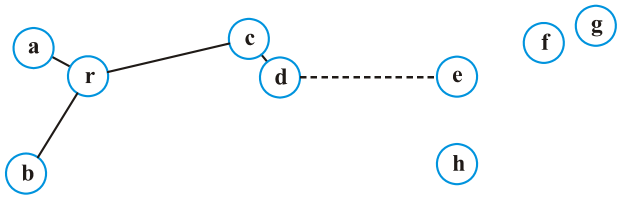


چهارمین یال انتخاب شده به صورت نقطه‌چین نشان داده شده است.

مثال ۳: با همان فرضیات مثال قبل چهارمین یالی که در الگوریتم پریم با شروع از رأس **a** انتخاب می‌شود کدام است؟

۱) (r,c) ۲) (c,d) ۳) (r,b) ۴) (d,e)

پاسخ: گزینه «۴» نحوه انتخاب یال‌ها در الگوریتم پریم به صورت زیر خواهد بود، که چهارمین یال انتخاب شده به صورت نقطه‌چین نشان داده شده است.



مثال ۴: با چه زمانی می‌توان تعیین کرد که یک یال مشخص مانند $e=(u,v)$ در یک درخت پوشای کمینه گراف داده شده با وزن‌های متمایز وجود دارد یا نه؟

۱) $O(V^2)$ ۲) $O(V+E^2)$ ۳) $O(V+E)$ ۴) $O(E\log V)$

پاسخ: گزینه «۳» در دو حالت می‌توان با قطعیت وجود یا عدم وجود یک یال مانند e در هر درخت پوشای کمینه را تعیین نمود:

اگر e یک یال برشی باشد آنگاه در هر درخت پوشای کمینه‌ای وجود دارد.

اگر e سنگین‌ترین یال موجود در یک دور باشد، آنگاه در هیچ درخت پوشای کمینه‌ای وجود ندارد. اما به صورت کلی‌تر می‌توان نکته زیر را بیان نمود:

نکته: یال $e=(u,v)$ به درخت MST تعلق ندارد اگر و تنها اگر بتوان با یال‌هایی که وزن تک تک آن‌ها از وزن e کمتر است از u به v رسید.

حال با دانستن این نکته کافیست تمام یال‌هایی که وزن آن‌ها بزرگتر یا مساوی وزن یال e می‌باشد را در زمان $O(V+E)$ از گراف حذف نمود و سپس دید که آیا در گراف باقیمانده مسیری از u به v وجود دارد یا خیر (با زمان $O(V+E)$).

مثال ۵: گراف ساده و همبند n رأسی با e یال G را درنظر بگیرید که وزن هر یال مقداری مثبت و یکتا باشد. در صورتی که به تمام یال‌ها، مقدار منفی M اضافه شود به طوری که باز هم وزن تمام یال‌ها مثبت بماند، چه تعداد از گزینه‌های زیر درست خواهند بود؟

الف) تعداد گام‌های کوتاه‌ترین مسیر بین دو رأس u و v کاهش نمی‌یابد. (منظور تعداد رأس‌های پیمایش شده برای رسیدن از u به v در مسیری است که کمترین وزن را داشته باشد.)

ب) درخت پوشای کمینه این گراف تغییر می‌کند.

ج) وزن درخت پوشای کمینه این گراف به مقدار $M(n-1)$ کاهش می‌یابد.

۱) 0 ۲) 1 ۳) 2 ۴) 3

پاسخ: گزینه «۳» عبارت «الف» صحیح است. در صورتی که یال با مقدار منفی نداشته باشیم، دور با طول منفی تشکیل نخواهد شد. در این شرایط کوتاه‌ترین مسیر از u به v شامل دور نمی‌شود. در صورتی که وزن تمام یال‌ها در شرایط ذکر شده کاهش یابد، مسیرهای انتخاب شده به شکلی تغییر می‌کنند که تعداد گام‌های کمتری نداشته باشند. فرض کنید d_1 مسیری با 5 گام از u به v با مجموع وزن 100 و d_2 مسیر دیگری با 7 گام و مجموع وزن 105 باشد. اگر از تمام یال‌ها مقدار 6 کم شود، وزن مسیر d_1 برابر 70 و وزن مسیر d_2 برابر 63 خواهد شد. در نتیجه مسیری با تعداد گام‌های مسیر بهینه کاهش می‌یابد.

ترتیب صعودی یال‌ها با توجه به وزنشان تغییری نمی‌کند. به همین دلیل، درخت پوشای کمینه گراف که با استفاده از الگوریتم کروسکال بدست می‌آید، تغییر نخواهد کرد. این درخت $n-1$ یال دارد که از هر یال آن مقدار M واحد کم شده است. در نتیجه وزن درخت پوشای کمینه گراف، به اندازه $M(n-1)$ واحد کاهش می‌یابد. در نتیجه عبارت «ب» غلط و «ج» صحیح است.

مثال ۶: گراف وزن‌دار G را درنظر بگیرید که وزن تمام یال‌های آن نامنفی و یکتا است. در مقدار وزن یال‌ها تغییری ایجاد می‌کنیم به این صورت که بجای مقداری مانند a، مقدار $f(a)$ را جایگزین می‌کنیم. به ازای کدام‌یک از توابع زیر، درخت پوشای کمینه گراف ممکن است تغییر کند؟

۱) $f(a)=2\log(a+3)$ ۲) $f(a)=(a-10)^2$ ۳) $f(a)=2\sqrt{a}+5$ ۴) $f(a)=3a^4-4$

پاسخ: گزینه «۲» تنها در صورتی که تابع $f(a)$ صعودی نباشد، درخت پوشای کمینه گراف، الزاماً ثابت نمی‌ماند. تنها تابعی که روی دامنه اعداد نامنفی (مقادیر وزن یال‌ها) صعودی نیست تابع $f(a)=(a-10)^2$ است. اگر وزن اولیه یک یال برابر 2 و وزن یال دیگر برابر 4 باشد، ترتیب این دو یال در بررسی شدن آن‌ها با الگوریتم کروسکال تغییر می‌کند.

مثال۷: **در چه تعداد از حالات زیر یال e در درخت پوشای کمینه گرافی که وزن تمام یال‌های آن متمایز است وجود دارد؟**

الف: یال e در هیچ دوری وجود نداشته باشد.

ب: یال e دومین کمترین وزن در گراف را داشته باشد.

ج: دوری وجود داشته باشد که یال e در آن کمترین وزن را داشته باشد.

۱) 0 ۲) 1 ۳) 2 ۴) 3

☑ **پاسخ:** گزینه «۳» حتی در صورتی که یال e روی دوری قرار داشته باشد که کم‌وزن‌ترین یال آن باشد، ممکن است هیچ یالی از آن دور در درخت پوشای کمینه قرار نگیرند. یالی که روی هیچ دوری قرار ندارد، می‌بایست در درخت پوشای کمینه قرار گیرد و دومین کوچکترین یال گراف، همواره توسط الگوریتم کروسکال انتخاب می‌گردد.

مثال۸: **چه تعداد از عبارت‌های زیر صحیح است؟**

(فرض کنید گراف G همبند است و یال e از این گراف بیشترین وزن در بین یال‌ها را دارد و این وزن یال در بین یال‌ها یکتاست.)

الف: یال e در هیچ‌یک از درخت‌های پوشای کمینه گراف G وجود ندارد.

ب: یال e در تمام درخت‌های پوشای کمینه گراف G وجود ندارد.

ج: در صورتی که گراف G دارای دور باشد، یال e در هیچ‌یک از درخت‌های پوشای کمینه این گراف وجود ندارد.

۱) 0 ۲) 1 ۳) 2 ۴) 3

☑ **پاسخ:** گزینه «۱» در صورتی که یال e یک پل (یال برشی) باشد، هر سه عبارت نادرست خواهند بود.

آزمون فصل سیزدهم

۱ـ کدام‌یک از گزینه‌های زیر درست است؟

۱) مسئله پیداکردن کوتاه‌ترین مسیر بین تمام جفت رئوس یک گراف با وزن‌های مثبت راه‌حل پویا دارد.

۲) مسئله پیداکردن طولانی‌ترین مسیر بین تمام جفت رئوس یک گراف با وزن‌های مثبت راه‌حل پویا دارد.

۳) مسئله پیدا کردن کوتاه‌ترین مسیر بین تمام جفت رئوس یک گراف راه‌حل پویا دارد اما مسئله طولانی‌ترین مسیر ندارد.

۴) مسئله پیدا کردن کوتاه‌ترین مسیر بین تمام جفت رئوس یک گراف راه‌حل پویا ندارد اما مسئله طولانی‌ترین مسیر دارد.

۲ـ کدام‌یک از الگوریتم‌های زیر جهت محاسبه کوتاه‌ترین مسیر بین تمام جفت رئوس یک گراف وزن‌دار قابل استفاده است؟

۱) kruskal　　۲) prim　　۳) Floyd-Warshall　　۴) solin

۳ـ میزان حافظه مصرفی در روش برنامه‌نویسی پویا برای مسئله فروشنده دوره‌گرد کدام است؟

۱) $\theta(n)$　　۲) $\theta(n^2)$　　۳) $\theta(n2^n)$　　۴) $\theta(n^2 2^n)$

۴ـ مرتبه زمانی الگوریتم پریم در صورتی که از ماتریس مجاورتی برای ذخیره‌سازی گراف استفاده شده باشد کدام است؟

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

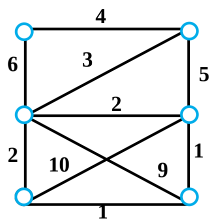
۵ـ از بین موارد زیر کدام موارد صحیح است؟

الف) الگوریتم پریم در گراف‌های چگال بهتر از الگوریتم کروسکال عمل می‌کند.

ب) یال‌های انتخاب شده در هر مرحله از الگوریتم پریم تشکیل یک درخت می‌دهند.

۱) فقط الف　　۲) فقط ب　　۳) الف و ب　　۴) هیچ‌کدام

۶ـ وزن درخت پوشای کمینه برای گراف زیر چقدر است؟



۱) 7

۲) 8

۳) 9

۴) 11

۷ـ کدام مورد صحیح است؟

الف) هر درخت پوشای کمینه یک درخت پوشای گلوگاه نیز می‌باشد.

ب) یال با کمترین وزن در یک گراف ساده در هر درخت پوشای کمینه وجود دارد.

ج) اگر وزن یال‌های گراف تکراری باشد، آنگاه بیش از یک درخت پوشای کمینه برای آن وجود دارد.

۱) الف و ب　　۲) ب و ج　　۳) هر سه مورد　　۴) فقط ب

۸ـ الگوریتم بلمن ـ فورد چند بار عمل relaxation را برای یک یال اجرا می‌کند؟

۱) 1 بار　　۲) $|V|-1$ بار　　۳) $|V^2|$ بار　　۴) $\log|V|$

۹ـ برای یافتن کوتاه‌ترین مسیر از یک رأس خاص به تمام رئوس دیگر در صورتی که وزن تمام یال‌ها برابر و مثبت باشد کدام الگوریتم مناسب‌تر است؟

۱) BFS　　۲) DFS　　۳) بلمن ـ فورد　　۴) دایکسترا

۱۰ـ در یک گراف که دارای یال‌هایی با وزن منفی می‌باشد، کدام جمله در مورد الگوریتم بلمن ـ فورد صحیح است؟

۱) همواره صحیح عمل می‌کند.

۲) ممکن است متوقف نشود.

۳) اگر گراف دور با وزن منفی نداشته باشد جواب صحیح را به دست می‌آورد.

۴) در حلقه بی‌نهایت می‌افتد.

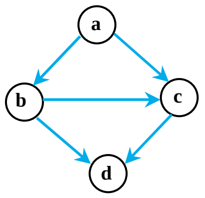
۱۱ـ مرتبه زمانی الگوریتم بلمن ـ فورد در یک گراف خلوت کدام است؟

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

۱۲ـ کدام‌یک از مسائل زیر را نمی‌توان در زمان $O(V+E)$ حل کرد؟

۱) یافتن مؤلفه‌های همبند قوی ۲) یافتن درخت پوشا ۳) مرتب‌سازی توپولوژیک ۴) یافتن MST

۱۳ـ در گراف زیر چند مسیر به طول 2 وجود دارد؟



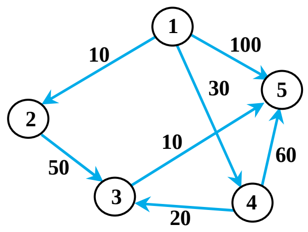
۱) 3

۲) 4

۳) 2

۴) 1

۱۴ـ در گراف زیر از گرهٔ 1 به 5 چند مسیر با طول 3 وجود دارد؟



۱) 1

۲) 2

۳) 3

۴) 4

۱۵ـ با در نظر گرفتن گراف فوق کمترین هزینه از گرهٔ 1 به 5 چند است؟

۱) 70 ۲) 90 ۳) 100 ۴) 60

بخش پنجم: مرتب‌سازی و مرتبه‌های آماری
فصل چهاردهم
«مرتب‌سازی‌های مقایسه‌ای»

تست‌های تألیفی فصل چهاردهم

مثال ۱: فرض کنید سیستمی در اختیار داریم که ادغام دو آرایه مرتب به طول‌های i و j را با پیچیدگی زمانی $\sqrt{i+j}$ محاسبه می‌کند. در این صورت پیچیدگی محاسباتی الگوریتم مرتب‌سازی ادغامی آرایه به‌طول n به چه شکل است؟

۱) $O(n)$ ۲) $O(\sqrt{n}\lg n)$ ۳) $O(\sqrt{n})$ ۴) $O(n\lg n)$

پاسخ: گزینه «۱» رابطه بازگشتی معادل با پیچیدگی محاسباتی این الگوریتم به‌صورت زیر است:

$$T(n)=2T(\frac{n}{2})+\sqrt{n}$$

پیچیدگی محاسباتی این رابطه از مرتبه $O(n)$ خواهد بود.

مثال ۲: سخت‌افزاری در اختیار داریم که می‌تواند k آرایه مرتب i عضوی را از مرتبه زمانی $\theta(i)$ با یکدیگر در یک آرایه مرتب ادغام نماید. با در اختیار داشتن این سخت‌افزار، مرتبه زمانی الگوریتم مرتب‌سازی ادغامی برای مرتب نمودن آرایه n عنصری به چه صورت خواهد بود؟

۱) $\theta(\frac{n\log n}{k\log k})$ ۲) $\theta(\frac{n}{k}\log\frac{n}{k})$ ۳) $\theta(\frac{n}{k}\log n)$ ۴) $\theta(n(\log n-\log k))$

پاسخ: گزینه «۱» رابطه زمان اجرای الگوریتم مرتب‌سازی ادغامی با استفاده از سخت‌افزار معرفی شده به شکل زیر است:

$$T(n)=kT(\frac{n}{k})+\theta(\frac{n}{k})\in\theta(\frac{n}{k}\log_k n)=\theta(\frac{n}{k}\times\frac{\log n}{\log k})=\theta(\frac{n\log n}{k\log k})$$

برای اثبات رابطه $\log_k n=\frac{\log n}{\log k}$ فرض کنید $d=\log_q p$ باشد که می‌توان $p=q^d$ را نتیجه گرفت. خواهیم داشت:

$$\frac{\log p}{\log q}=\frac{\log_c p}{\log_c q}=\frac{\log_c q^d}{\log_c q}=\frac{d\log_c q}{\log_c q}=d=\log_q p$$

مثال ۳: سامانه‌ای در اختیار داریم که با دریافت یک آرایه به طول n از اعداد متمایز، زیرآرایه‌ای به طول $\frac{n}{2}$ را باز می‌گرداند که تمام عناصر آن حداقل از $\frac{1}{8}$ عناصر آرایه بزرگتر و از $\frac{1}{8}$ عناصر آرایه کوچکتر هستند. اگر مرتبه زمانی بازگردانی خروجی این سامانه از مرتبه $\theta(\log n)$ باشد، با کمک گرفتن از این سامانه مرتبه زمانی مرتب‌سازی سریع در حالت متوسط از چه مرتبه‌ای خواهد بود؟

۱) $\theta(n)$ ۲) $\theta(n\sqrt{\log n})$ ۳) $\theta(n\log\log n)$ ۴) $\theta(n\log n)$

پاسخ: گزینه «۴» تنها کارایی این سامانه این است که می‌توان هریک از اعداد زیرآرایه را به‌عنوان عنصر محور در مرتب‌سازی سریع به‌کار گرفت. در این حالت مرتبه زمانی الگوریتم مرتب‌سازی سریع در حالت متوسط و بدترین حالت، همواره $\theta(n\log n)$ خواهد بود. رابطه بازگشتی معادل با الگوریتم به شکل زیر است:

$$T(n)=T(c_1n-1)+T(c_2n)+\theta(n)+\theta(\log n)\quad,\quad c_1+c_2=1\quad,\quad c_1,c_2\geq\frac{1}{8}$$

مثال ۴: سخت‌افزاری در اختیار داریم که با دریافت یک آرایه نامرتب به طول n و عدد طبیعی k در بازه $1 \le k \le n$، kامین کوچکترین عنصر این آرایه را از مرتبه $\theta(\log n)$ بازمی‌گرداند. در این صورت از چه مرتبه‌ای می‌توان این آرایه را مرتب نمود؟

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

پاسخ: گزینه «۲» می‌توان $\frac{n}{\log n}-1$ مرتبه از سخت‌افزار مذکور برای یافتن $\log n$امین عنصر، $2\log n$امین عنصر، ... و $(\frac{n}{\log n}-1)\log n$امین عنصر استفاده نمود. سپس با یک مرتبه بررسی آرایه، آن را به $\frac{n}{\log n}$ قسمت تقسیم نمود که تمام اعداد موجود در هر قسمت، بین دو عدد استخراج شده با سخت‌افزار فوق و بزرگتر از تمام اعداد موجود در قسمت‌های قبل باشد. پس از مرتب‌سازی این قسمت‌ها، آرایه کلی مرتب خواهد شد. رابطه بازگشتی این الگوریتم به شکل زیر خواهد بود.

$$T(n)=\frac{n}{\log n}T(\log n)+\theta(n)$$

مرتبه زمانی این الگوریتم را با استفاده از تکرار محاسبه می‌کنیم. شرط پایان در حالتی است که مقدار پارامتر تابع، کمتر از عدد ثابتی مانند 2 باشد. $(T(n)=1; n\le 2)$ خواهیم داشت:

$$T(n)=\frac{n}{\log n}T(\log n)+\theta(n)=\frac{n}{\log n}(\frac{\log n}{\log\log n}T(\log\log n)+\theta(\log n))+\theta(n)$$
$$=\frac{nT(\log\log n)}{\log\log n}+2\theta(n)=\cdots\simeq\frac{nT(1)}{1}+\theta(n)\log^* n\in\theta(n\log^* n)$$

مثال ۵: آرایه A حاوی n عنصر متمایز در اختیار داریم. برای مرتب‌سازی این آرایه، دو اندیس متمایز i و j را انتخاب می‌کنیم. درصورتی که $i<j$ و $A[i]>A[j]$ برقرار باشد، این دو عنصر آرایه را عوض می‌کنیم. برای مرتب‌سازی این آرایه، تعداد جابجایی‌ها از چه مرتبه‌ای است؟

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

پاسخ: گزینه «۲» حداقل تعداد مقایسه‌ها از مرتبه $O(n^2)$ است. برای نمونه می‌توان تعداد جابجایی‌ها در مرتب‌سازی درجی را درنظر گرفت. در آرایه اولیه، هر عدد حداکثر $O(n)$ واحد با مکان واقعی‌اش فاصله دارد. در نتیجه مجموع مقادیر فاصله هر عدد تا جایگاه واقعی‌اش از مرتبه $O(n^2)$ است. با هر جابجایی، این مقدار کاهش می‌یابد. زیرا مجموع اختلاف دو عدد جابجا شده از جایگاه واقعی‌شان به اندازه $2(j-i)$ واحد کاهش خواهد یافت. زمانی آرایه مرتب می‌شود که مجموع اختلاف‌ها برای آرایه برابر 0 شود. درنتیجه با حداکثر $O(n^2)$ جابجایی می‌توان آرایه را مرتب نمود. توجه کنید که این معیار برای جابجایی‌ها است نه برای مقایسه‌ها.

مثال ۶: تابع F(A,i) در اختیار داریم که با گرفتن آرایه A و اندیس i، عناصر i تا i+k از آرایه A را به‌صورت صعودی مرتب می‌کند. کدام‌یک از دو الگوریتم زیر عمل مرتب‌سازی را به درستی انجام می‌دهد؟

الف: به ازای i از 1 تا n-k تابع F(A,i) را فراخوانی می‌کنیم. سپس به ازای i از n-k تا 1 تابع F(A,i) را فراخوانی می‌کنیم.

ب: به تعداد $\frac{n}{k}$ مرتبه (به ازای j از 1 تا $\frac{n}{k}$)، به ازای i از 1 تا $\frac{n}{k}$ تابع F(A,i) را فراخوانی می‌کنیم.

۱) درست، درست ۲) درست، نادرست ۳) نادرست، درست ۴) نادرست، نادرست

پاسخ: گزینه «۴» الگوریتم «الف» به درستی کار نمی‌کند. زیرا اگر آرایه به ترتیب نزولی مرتب شده باشد، در مسیر صعودی i، k بزرگترین عضو به جایگاهشان منتقل می‌شوند و در مسیر نزولی i، k کوچکترین عضو در مکان خود قرار می‌گیرند. در این شرایط، عناصر میانی در جای صحیح خود قرار نمی‌گیرند. الگوریتم «ب» به درستی کار نمی‌کند. این الگوریتم در تکرار اول، پنجره‌های متوالی به طول $\frac{k}{n}$ را مرتب می‌کند. در سایر تکرارها، فقط در همان بازه‌ها مرتب‌سازی را تکرار می‌کند. خروجی نهایی با خروجی پس از 1 تکرار برابر است. اگر الگوریتم به تعداد $\frac{2n}{k}$ مرتبه، به ازای i از 1 تا $\frac{2n}{k}$ تابع F(A,i) فراخوانی می‌شد، عمل مرتب‌سازی به درستی انجام می‌گرفت.

مثال۷: **ترتیب عناصر $4n+3$ عضوی A (با عناصر متمایز) را عوض می‌کنیم و در آرایه B قرار می‌دهیم؛ بطوری که دنباله حاصل در آرایه B به‌صورت زیر باشد.**

$$b_1 < b_2 > b_3 > b_4 < b_5 < b_6 > b_7 > b_8 < \cdots$$

یعنی به ازای اعداد صحیح k، عناصر در جایگاه‌های $2(2k+1)$ از دو عنصر مجاورشان بزرگتر و عناصر در جایگاه‌های $4k$ از دو عنصر مجاورشان کوچکتر باشند. بدست آوردن آرایه‌ای به این شکل از چه مرتبه‌ای ممکن است؟

۱) $\theta(n)$　　۲) $\theta(n \log\log n)$　　۳) $\theta(n \log n)$　　۴) $\theta(n\sqrt{n})$

☑ **پاسخ:** گزینه «۱» ابتدا آرایه B را برابر آرایه A درنظر می‌گیریم. سپس به ازای هر جایگاه زوج مانند $2u$ از 2 تا $4n+2$ محاسبات زیر را انجام می‌دهیم:

اگر $2u$ در جایگاه $2(2k+1)$ بود، (بیشینه محلی) ابتدا آن را با بیشینه عناصر $< 2u-2, 2u-1, 2u >$ جابجا می‌کنیم تا عنصر بیشینه بین این سه عنصر باشد. سپس آن را با بیشینه عناصر $< 2u, 2u+1, 2u+2 >$ جابجا می‌کنیم تا عنصر بیشینه بین این سه عنصر نیز باشد.

اگر $2u$ در جایگاه $4k$ بود، (کمینه محلی) ابتدا آن را با کمینه عناصر $< 2u-2, 2u-1, 2u >$ جابجا می‌کنیم تا عنصر کمینه بین این سه عنصر باشد. سپس آن را با کمینه عناصر $< 2u, 2u+1, 2u+2 >$ جابجا می‌کنیم تا عنصر کمینه بین این سه عنصر نیز باشد.

با یک مرتبه بررسی آرایه، به شکل مطلوب خواهیم رسید. مرتبه زمانی این کار به‌صورت $\theta(n)$ خواهد بود.

مثال۸: **فرض کنید داده‌های مورد استفاده برای مرتب‌سازی هرمی با فواصل زمانی نسبتاً زیاد به ترتیب از ورودی وارد می‌شوند و در نتیجه برای ساخت heap به ترتیب داده‌های وارد شده را در یک heap تهی درج می‌کنیم. در این صورت مرتبه زمانی ساخت درخت کدام است؟ (فقط زمانی که صرف ساخت درخت می‌شود)**

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

☑ **پاسخ:** گزینه «۱» اگر تمام داده‌ها از قبل وجود داشته باشند، آنگاه مرتبه زمانی $\theta(n)$ خواهد بود اما اگر داده‌ها به ترتیب وارد شوند مرتبه زمانی ساخت درخت برابر $O(n \log n)$ می‌باشد.

آزمون فصل چهاردهم

۱ـ مرتبه زمانی کدام‌یک از مرتب‌سازی‌های زیر در بهترین حالت $\theta(n)$ است؟

۱) مرتب‌سازی ادغامی ۲) مرتب‌سازی حبابی ۳) مرتب‌سازی سریع ۴) مرتب‌سازی درجی

۳ـ مقدار فضای کمکی مصرفی در الگوریتم heap sort کدام است؟

۱) $\theta(n)$ ۲) $\theta(\log n)$ ۳) $\theta(1)$ ۴) $\theta(n\log n)$

۳ـ میانگین تعداد وارونگی‌ها در یک لیست n عنصری کدام است؟

۱) $\frac{n(n-1)}{2}$ ۲) $\frac{n(n-1)}{4}$ ۳) n^2 ۴) $\frac{3n^2}{2}$

۴ـ اگر یک آرایه n عنصری به وسیله الگوریتم مرتب‌سازی ادغامی مرتب شود، آنگاه در بدترین حالت چه تعداد مقایسه انجام می‌شود (فرض کنید n توانی از 2 باشد)؟

۱) $n\log_2^n$ ۲) $n\log_2^n - n + 1$ ۳) $n\log_2^n + n - 1$ ۴) $n\log_2^n + n + 1$

۵ـ کدام مرتب‌سازی زیر هیچ‌گاه یک عنصر را با تمام عناصر دیگر لیست مقایسه نمی‌کند؟

۱) حبابی ۲) انتخابی ۳) درجی ۴) هرمی

۶ـ با چه مرتبه زمانی می‌توان تعیین کرد که رابطه $x_i + x_j \ge x_k$ برای تمام عناصر آرایه زیر برقرار است یا نه؟

x_1	x_2	x_3	......	x_n

۱) $\theta(n)$ ۲) $\theta(\log_2^n)$ ۳) $\theta(1)$ ۴) $\theta(n\log n)$

۷ـ اگر A_1 و A_2 دو آرایه مرتب باشند، آنگاه زمان یافتن $A_1 \cup A_2$ کدام است؟ (تعداد عناصر هر دو آرایه برابر n می‌باشد)

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

۸ـ اگر A_1 و A_2 دو لیست نامرتب هر کدام با n عنصر باشند، آنگاه زمان یافتن $A_1 \cap A_2$ کدام است؟

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

۹ـ حداقل تعداد مقایسه‌ها در جست‌وجوهای مبتنی بر مقایسه در بدترین حالت برای مرتب‌سازی یک لیست n عنصری چقدر است؟

۱) $\log n$ ۲) $\log n!$ ۳) n^2 ۴) n

۱۰ـ برای مرتب‌سازی یک آرایهٔ N تایی به روش Bubble – Sort حداکثر چند تعویض لازم است؟

۱) $\frac{n(n-1)}{2}$ ۲) n ۳) $\text{Log}_2(n-1)$ ۴) $\frac{n}{2}$

۱۱ـ رویه‌ی زیر کدام روش مرتب‌سازی است؟

```
void sort(elementlist a,int n)
    {int i,j,v;
    for(j=2;j<=n;++j)
    {v=a[i];j=i;
    while(a[j-1]> v)
    {a[j]=a[j-1];
    j=j-1;
    }
a[j]=v;
}
}
```

۱) مرتب‌سازی حبابی

۲) مرتب‌سازی انتخابی

۳) مرتب‌سازی درجی

۴) مرتب‌سازی تعویضی

۱۲ـ از کدام الگوریتم مرتب‌سازی، برای مرتب کردن عناصر درون یک فایل استفاده می‌کنید؟

۱) Bubble ۲) Insertion ۳) Quick ۴) Merge

۱۳ـ الگوریتم **Insertion Sort** یک رشتهٔ **n** تایی را در بدترین حالت با چه سرعتی مرتب می‌کند؟

۱) O(n) ۲) $O(n^2)$ ۳) O(Logn) ۴) O(nLogn)

۱۴ـ الگوریتم **Heap Sort** یک رشتهٔ **n** تایی را در بدترین حالت با چه سرعتی مرتب می‌کند؟

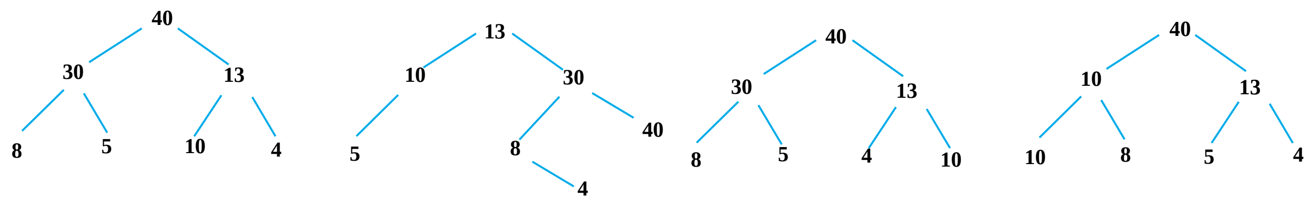
۱) O(n) ۲) $O(n^2)$ ۳) O(Logn) ۴) O(nLogn)

۱۵ـ برای مرتب‌سازی یک آرایهٔ **n** تایی به روش **Selection Sort** حداکثر چند تعویض لازم است؟

۱) $\frac{n(n-1)}{2}$ ۲) $n-1$ ۳) $Log_2(n-1)$ ۴) nLogn

۱۶ـ فرض کنید آرایهٔ زیر را با روش **heap Sort** می‌خواهیم مرتب کنیم، پس از اجرای مرحلهٔ اوّل الگوریتم درخت **heap** حاصل شده کدام است؟

30	40	10	8	5	4	13



(۱) (۲) (۳) (۴)

۱۷ـ در روش **Selection Sort** برای مرتب‌سازی یک آرایهٔ **n** تایی که به صورت صعودی مرتب باشد، چند مقایسه لازم است؟

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

۱۸ـ **k** لیست پیوندی مرتب با مجموع **n** عنصر داده شده است و تعداد عناصر هر لیست پیوندی مشخص نیست. کدام‌یک از گزینه‌های زیر مرتبهٔ سریع‌ترین الگوریتم برای ادغام این لیست‌های پیوندی است؟

۱) $\theta(n\log k)$ ۲) $\theta(nk)$ ۳) $\theta(n+k)$ ۴) $\theta(n\log n)$

۱۹ـ تعداد وارونگی‌های آرایهٔ $[2,4,6,8,10,12,14,16,18,20]$ کدام است؟

۱) 0 ۲) 1 ۳) 9 ۴) 45

۲۰ـ اگر در الگوریتم مرتب‌سازی ادغامی، به جای تقسیم آرایه به دو بخش مساوی، آرایه را هر بار از یک مکان تصادفی تقسیم کنیم، میانگین زمان اجرای این الگوریتم چه تغییری می‌کند؟

۱) از حالت معمول کندتر می‌شود.
۲) از حالت معمول سریع‌تر می‌شود.
۳) O(n) می‌شود.
۴) مانند حالت معمول خواهد بود.

فصل پانزدهم
«مرتب‌سازی‌های غیرمقایسه‌ای»

تست‌های تألیفی فصل پانزدهم

مثال ۱: با فرض اینکه n داده صحیح با مقادیر 1 تا m در اختیار داشته باشیم، مرتب‌سازی شمارشی در چه صورت بهتر از مرتب‌سازی ادغامی عمل می‌کند؟ (بهترین کران بالای ممکن را انتخاب کنید.)

۱) $m \in o(n)$ ۲) $m \in o(\log n)$ ۳) $m \in o(2^n)$ ۴) $m \in o(n \log n)$

☑ **پاسخ:** گزینه «۴» مرتب‌سازی ادغامی از مرتبه $\theta(n \log n)$ و مرتب‌سازی شمارشی از مرتبه $\theta(m+n)$ این آرایه را مرتب می‌کنند. برای اینکه مرتبه زمانی مرتب‌سازی شمارشی کمتر از مرتب‌سازی ادغامی باشد، می‌بایست رابطه $m \in o(n \log n)$ برقرار شود.

مثال ۲: اگر مبنای مورد استفاده در مرتب‌سازی مبنایی برابر n باشد و بزرگترین عدد استفاده شده $n^3 - 1$ باشد، آنگاه مرتبه زمانی الگوریتم کدام است؟

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

☑ **پاسخ:** گزینه «۳» دقت کنید که عدد $n^3 - 1$ در مبنای n یک عدد 3 رقمی می‌باشد بنابراین مرتبه زمانی الگوریتم عبارت است از:

$\theta(d(n+r)) = \theta(3(n+n)) = \theta(n)$

مثال ۳: آرایه‌ای متشکل از n عدد حقیقی در بازه $[-n\lg\lg n, n\lg\lg n]$ در اختیار داریم. مرتب‌سازی این آرایه از چه مرتبه‌ای خواهد بود؟

۱) $\theta(n)$ ۲) $\theta(n \log\log n)$ ۳) $\theta(\frac{n \log n}{\log\log n})$ ۴) $\theta(n \log n)$

☑ **پاسخ:** گزینه «۴» با توجه به اینکه اعداد آرایه مقدار حقیقی دارند، با استفاده از مرتب‌سازی شمارشی قادر به مرتب‌سازی آن‌ها نخواهیم بود. مرتب‌سازی این آرایه از مرتبه $\theta(n \log n)$ خواهد بود.

آزمون فصل پانزدهم

۱ـ اگر یک لیست n عنصری شامل اعداد صحیح در بازه $[0,n^2]$ باشد، آنگاه در حالت کلی با چه زمانی می‌توان آن را مرتب نمود؟

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

۲ـ کدام‌یک از مرتب‌سازی‌های زیر stable می‌باشد؟

۱) مرتب‌سازی سریع ۲) مرتب‌سازی شمارشی ۳) مرتب‌سازی درختی ۴) مرتب‌سازی انتخابی

۳ـ کدام‌یک از الگوریتم‌های مرتب‌سازی زیر inplace نمی‌باشند؟

۱) مرتب‌سازی انتخابی ۲) مرتب‌سازی درجی ۳) مرتب‌سازی حبابی ۴) مرتب‌سازی شمارشی

۴ـ اعداد مقابل را در نظر بگیرید: **321 ,129 ,703 , 340 , 809 , 267**

اگر از الگوریتم مرتب‌سازی مبنایی (Radix) استفاده شود در پایان مرحله دوم اعداد فوق به چه صورت قرار می‌گیرند؟

۱) 129 , 321 ,340 , 267 , 809 , 703
۲) 809 ,703 ,129 , 321, 267 ,340
۳) 703 ,809 ,321 ,129 , 340, 267
۴) 129 ,340 ,321, 267 , 703 ,809

۵ـ n عدد صحیح k رقمی داریم (k ثابت و مستقل از n). کدام گزینه درست‌تر است؟

۱) مرتب‌سازی این اعداد در زمان $\theta(n)$ به حافظهٔ $\Omega(n^2)$ احتیاج دارد.
۲) این اعداد را می‌توان در زمان $\theta(n)$ و با حافظهٔ $O(n)$ مرتب کرد.
۳) مرتب‌سازی این اعداد دست‌کم زمان $\Omega(n\log n)$ و حافظهٔ $\Omega(n)$ نیاز دارد.
۴) این اعداد را می‌توان در زمان $\theta(n\log n)$ و با حافظهٔ $\theta(n)$ مرتب کرد.

فصل شانزدهم

«مسأله انتخاب (Selection)»

تست‌های تألیفی فصل شانزدهم

مثال ۱: اگر هدف مرتب کردن $\sqrt{n}$ عنصر از کوچکترین عناصر یک آرایه باشد، با چه مرتبه زمانی می‌توان این کار را انجام داد؟

۱) $\theta(\sqrt{n}\log n)$　　۲) $\theta(n\log n)$　　۳) $\theta(n)$　　۴) $\theta(\log_n^2)$

☑ **پاسخ:** گزینه «۱» ابتدا $\sqrt{n}$ امین کوچکترین عنصر آرایه را در زمان $O(n)$ می‌یابیم و سپس با استفاده از الگوریتم partition تمام $\sqrt{n}$ کوچکترین عناصر آرایه را به قبل آن منتقل می‌کنیم، حال کافی است با یک الگوریتم مرتب‌سازی این زیر آرایه با اندازه $\sqrt{n}$ را مرتب کنیم که دارای زمان $\theta(\sqrt{n}\log\sqrt{n}) = \theta(\sqrt{n}\log n)$ می‌باشد. بنابراین در کل مرتبه زمانی $\theta(\sqrt{n}\log n)$ خواهد بود.

مثال ۲: فرض کنید یک آرایه مرتب در اثر شیفت چرخشی عناصر به سمت راست به یک آرایه نامرتب تبدیل شده است. با چه مرتبه زمانی می‌توان بزرگترین عنصر آرایه را یافت؟

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

☑ **پاسخ:** گزینه «۳» برای یافتن بزرگترین یا کوچکترین عنصر آرایه، باید مرزی را بیابیم که کوچکترین و بزرگترین عنصر با یکدیگر همسایه می‌باشند. برای انجام این کار عناصر x_1، x_2 و x_3 را به صورت زیر در نظر می‌گیریم:

$\overset{\longleftrightarrow}{\frac{n}{4}}$	x_1	$\overset{\longleftrightarrow}{\frac{n}{4}}$	x_2	$\overset{\longleftrightarrow}{\frac{n}{4}}$	x_3	$\overset{\longleftrightarrow}{\frac{n}{4}}$

حال اگر برای یکی از xها رابطه زیر برقرار نباشد، آنگاه مرز موردنظر به دست آمده است:

عنصر بعد از x_i در آرایه $\le x_i \le$ عنصر قبل از x_i در آرایه

در غیر این صورت اگر $x_1 \le x_2 \le x_3$، آنگاه می‌توان عناصر بین x_1 تا x_3 را حذف کرد و مجدداً همین روند را تکرار نمود. اما در صورتی که $x_2 < x_1$ باشد مرز مورد نظر بین x_1 و x_2 است و اگر $x_3 < x_2$ باشد، مرز موردنظر بین x_2 و x_3 است. در نتیجه در هر مرحله حداقل $\frac{n}{2}$ از عناصر برای بررسی بیشتر حذف می‌شوند بنابراین، مرتبه زمانی الگوریتم $O(\log n)$ خواهد بود.

مثال ۳: آرایه A به طول n در اختیار داریم که k امین عنصر از آن، عنصر بیشینه است. همچنین به ازای هر i نامنفی در مقادیر مجاز از اندیس آرایه، دو رابطه $A[k-i] \ge A[k-i-1]$ و $A[k+i] \ge A[k+i+1]$ برقرار هستند. در این صورت، در دو حالتی که مقادیر آرایه متمایز باشند یا بتوانند مشابه باشند، مقدار k را می‌توان از چه مرتبه‌ای تعیین نمود؟

۱) در هر دو حالت $O(\log n)$.

۲) حالت متمایز بودن اعداد $O(\log n)$ و حالت مجاز بودن تکرار $O(\log n\log\log n)$.

۳) حالت متمایز بودن اعداد $O(\log n)$ و حالت مجاز بودن تکرار $O(n)$.

۴) حالت متمایز بودن اعداد $O(\log n\log\log n)$ و حالت مجاز بودن تکرار $O(n)$.

☑ **پاسخ:** گزینه «۳» در صورتی که همه اعداد متمایز باشند، از روشی مشابه با جست‌وجوی دودویی استفاده می‌کنیم. ابتدا عنصر میانی آرایه را بررسی می‌کنیم. این عنصر را با عنصر قبل و بعد از آن مقایسه می‌کنیم. با توجه به وضعیت دنباله مسأله، این عدد حداقل از یکی از دو عضو مجاورش بزرگتر خواهد بود. در صورتی که از هر دو عضو مجاورش بزرگتر باشد، عنصر k ام را یافته‌ایم. در غیر این صورت اگر عنصر میانی از عنصر سمت چپ خود بزرگتر باشد، زیردنباله سمت راست را بررسی می‌کنیم و در غیر این صورت، زیردنباله سمت چپ را بررسی خواهیم کرد. در نتیجه، آرایه با 2 جست‌وجو، نصف خواهد شد. رابطه بازگشتی معادل با این جست‌وجو به صورت $T(n) = T(\frac{n}{2}) + \theta(1)$ خواهد بود که از مرتبه $O(\log n)$ است. در صورتی که اعداد دنباله بتوانند مشابه باشند، این شرایط را نخواهیم داشت. دنباله‌ای با n-1 مقدار 0 و یک مقدار 1 درنظر بگیرید. این دنباله شرط ورودی مسأله را دارد و برای یافتن عنصر با مقدار 1 باید آرایه را به طور کامل بررسی کنیم. در این شرایط مرتبه یافتن عنصر بیشینه از مرتبه $O(n)$ خواهد بود.

مثال ۴: یافتن سومین کوچکترین عنصر در مجموعه‌ای شامل n عدد متمایز به چه تعداد مقایسه نیاز دارد؟ نزدیک‌ترین گزینه به جواب را انتخاب نمایید.

۱) $3n$	۲) $n+\log n+\log\log n$	۳) $n+3\log n$	۴) $n+2\log n+\log\log n$

پاسخ: گزینه «۴» با $n-1$ مقایسه، کوچکترین عنصر مجموعه را می‌توان پیدا کرد. در این مقایسه‌ها ابتدا باید یک ساختار درختی را درنظر گرفت که دو عنصر که با i عنصر دیگر مقایسه شدند و کوچکتر بودند، باهم مقایسه شوند. در این صورت عنصر کمینه با دقیقاً $\log n$ عنصر مقایسه می‌شود. دومین کوچکترین عنصر یکی از عناصری است که با عنصر کمینه مقایسه شده است.

حال به $\log n-1$ مقایسه برای تعیین دومین کوچکترین عنصر از میان مجموعه عناصری که با عنصر کمینه مقایسه شده‌اند نیاز داریم. سومین کوچکترین عنصر را می‌بایست از بین عناصری که با دومین کوچکترین عنصر (در هر دو جست‌وجو برای یافتن کوچکترین و دومین کوچکترین عنصر) مقایسه شده‌اند تعیین نمود. دومین کوچکترین عنصر برای جست‌وجوی کوچکترین عنصر، حداکثر با $\log n-1$ عنصر و برای جست‌وجوی دومین کوچکترین عنصر، با $\log\log n$ عنصر مقایسه شده است.

حال حداکثر $\log n+\log\log n-1$ عنصر داریم که می‌بایست کوچکترین عنصر از بین آن‌ها را پیدا کنیم. تعداد مقایسات در این مجموعه برابر $\log n+\log\log n-2$ است. تعداد کل مقایسات برابر مجموع حالت‌های فوق یعنی $n+2\log n+\log\log n-3$ خواهد بود.

مثال ۵: در یک آرایه n عضوی نامرتب A به دنبال یافتن دو اندیس i و j هستیم به طوری که مقدار $A[j]-A[i]$ بیشینه شود و روابط $1\le i<j\le n$ و $j-i\le\log n$ برقرار باشد. یافتن این دو اندیس از چه مرتبه‌ای ممکن است؟

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

پاسخ: گزینه «۴» می‌توان به ازای j از 2 تا n، مقدار عنصر مینیمم از بین (حداکثر) $\log n$ عنصر قبل از j را یافت این مقدار را از $A[j]$ کم نمود. در صورتی که این مقدار از مقدار بیشینه بدست آمده بیشتر باشد، مقدار j و اندیس عنصر کمینه جایگزین مقادیر پیشین می‌گردد. یافتن عنصر کمینه در بازه‌های این مسأله از مرتبه $O(n)$ است. دو اندیس i و j را می‌توان از مرتبه $O(n)$ و با یک مرتبه بررسی آرایه تعیین نمود.

آزمون فصل شانزدهم

۱ـ انتخاب k امین کوچکترین عنصر یک لیست مرتب در چه زمانی قابل انجام است؟

۱) $\theta(1)$　　۲) $\theta(n)$　　۳) $\theta(\log n)$　　۴) $\theta(n\log n)$

۲ـ تعداد مقایسه‌های لازم برای یافتن همزمان min و max در یک آرایه n عنصری کدام است؟ (فرض کنید n یک عدد فرد باشد)

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

۳ـ مقدار حافظه کمکی موردنیاز در الگوریتم افراز کدام است؟

۱) $\theta(n)$　　۲) $\theta(1)$　　۳) $\theta(\log n)$　　۴) $\theta(n^2)$

۴ـ یافتن k امین کوچک‌ترین عنصر بین n عنصر با استفاده از مرتب‌سازی سریع در بدترین حالت در چه مرتبه‌ای انجام می‌شود؟

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

۵ـ دومین کوچک‌ترین عنصر بین n عنصر را با چند مقایسه می‌توان به دست آورد؟

۱) $n+\lfloor \log n \rfloor-1$　　۲) $n+\lfloor \log n \rfloor-2$　　۳) $n+\lceil \log n \rceil-2$　　۴) $n+\lceil \log n \rceil-1$

پاسخنامه آزمون‌ها

فصل اول: پیچیدگی زمانی الگوریتم‌ها

۱ـ گزینه «۲»	۲ـ گزینه «۴»	۳ـ گزینه «۲»	۴ـ گزینه «۳»	۵ـ گزینه «۱»
۶ـ گزینه «۳»	۷ـ گزینه «۱»	۸ـ گزینه «۴»	۹ـ گزینه «۴»	۱۰ـ گزینه «۴»
۱۱ـ گزینه «۳»	۱۲ـ گزینه «۳»	۱۳ـ گزینه «۲»	۱۴ـ گزینه «۴»	۱۵ـ گزینه «۲»
۱۶ـ گزینه «۲»	۱۷ـ گزینه «۴»	۱۸ـ گزینه «۱»	۱۹ـ گزینه «۱»	۲۰ـ گزینه «۳»
۲۱ـ گزینه «۲»	۲۲ـ گزینه «۳»	۲۳ـ گزینه «۴»	۲۴ـ گزینه «۱»	۲۵ ـ گزینه «۳»
۲۶ـ گزینه «۳»	۲۷ـ گزینه «۲»	۲۸ـ گزینه «۴»	۲۹ـ گزینه «۲»	۳۰ـ گزینه «۲»
۳۱ـ گزینه «۴»	۳۲ـ گزینه «۴»	۳۳ـ گزینه «۲»	۳۴ـ گزینه «۱»	۳۵ـ گزینه «۳»
۳۶ـ گزینه «۲»	۳۷ـ گزینه «۴»	۳۸ـ گزینه «۱»	۳۹ـ گزینه «۲»	۴۰ـ گزینه «۳»

فصل دوم: آنالیز سرشکن

۱ـ گزینه «۱»	۲ـ گزینه «۳»	۳ـ گزینه «۳»	۴ـ گزینه «۱»	۵ـ گزینه «۴»

فصل سوم: مقدمه‌ای بر پیچیدگی محاسباتی (Computational Complexity)

۱ـ گزینه «۲»	۲ـ گزینه «۲»	۳ـ گزینه «۱»	۴ـ گزینه «۴»	۵ـ گزینه «۴»
۶ـ گزینه «۱»	۷ـ گزینه «۳»	۸ـ گزینه «۳»	۹ـ گزینه «۱»	۱۰ـ گزینه «۱»

فصل چهارم: داده ساختارهای مقدماتی

۱ـ گزینه «۴»	۲ـ گزینه «۲»	۳ـ گزینه «۳»	۴ـ گزینه «۳»	۵ـ گزینه «۲»
۶ـ گزینه «۱»	۷ـ گزینه «۱»	۸ـ گزینه «۳»	۹ـ گزینه «۲»	۱۰ـ گزینه «۴»

فصل پنجم: داده ساختارهای مبتنی بر گراف

۱ـ گزینه «۱»	۲ـ گزینه «۱»	۳ـ گزینه «۴»	۴ـ گزینه «۳»	۵ـ گزینه «۱»
۶ـ گزینه «۴»	۷ـ گزینه «۲»	۸ـ گزینه «۲»	۹ـ گزینه «۲»	۱۰ـ گزینه «۱»
۱۱ـ گزینه «۱»	۱۲ـ گزینه «۲»	۱۳ـ گزینه «۳»	۱۴ـ گزینه «۴»	۱۵ ـ گزینه «۳»
۱۶ـ گزینه «۴»	۱۷ـ گزینه «۴»	۱۸ـ گزینه «۲»	۱۹ـ گزینه «۳»	۲۰ـ گزینه «۴»

فصل ششم: داده ساختارهای پیشرفته

۱ـ گزینه «۳»	۲ـ گزینه «۲»	۳ـ گزینه «۴»	۴ـ گزینه «۱»	۵ـ گزینه «۳»
۶ـ گزینه «۲»	۷ـ گزینه «۳»	۸ـ گزینه «۱»	۹ـ گزینه «۱»	۱۰ـ گزینه «۴»

فصل هفتم: درهم‌سازی

۱ـ گزینه «۳»	۲ـ گزینه «۱»	۳ـ گزینه «۲»	۴ـ گزینه «۲»	۵ـ گزینه «۴»
۶ـ گزینه «۲»	۷ـ گزینه «۴»	۸ـ گزینه «۱»	۹ـ گزینه «۳»	۱۰ـ گزینه «۱»

فصل هشتم: الگوریتم‌های تقسیم و غلبه (Divide and Conquer)

۱ـ گزینه «۴»	۲ـ گزینه «۱»	۳ـ گزینه «۳»	۴ـ گزینه «۴»	۵ـ گزینه «۱»
۶ـ گزینه «۴»	۷ـ گزینه «۱»	۸ـ گزینه «۳»	۹ـ گزینه «۴»	۱۰ـ گزینه «۲»

فصل نهم: برنامه‌ریزی پویا (Dynamic Programming)

۱ـ گزینه «۳»	۲ـ گزینه «۳»	۳ـ گزینه «۳»	۴ـ گزینه «۳»	۵ـ گزینه «۱»
۶ـ گزینه «۱»	۷ـ گزینه «۲»	۸ـ گزینه «۴»	۹ـ گزینه «۳»	۱۰ـ گزینه «۳»

فصل دهم: الگوریتم‌های حریصانه (Greedy Algorithms)

۱ـ گزینه «۲»	۲ـ گزینه «۳»	۳ـ گزینه «۱»	۴ـ گزینه «۳»	۵ـ گزینه «۴»

فصل یازدهم: الگوریتم‌های مبتنی بر جست‌وجوی درخت فضای حالت

۱ـ گزینه «۳»	۲ـ گزینه «۱»	۳ـ گزینه «۳»	۴ـ گزینه «۴»	۵ـ گزینه «۲»
۶ـ گزینه «۳»	۷ـ گزینه «۱»	۸ـ گزینه «۲»	۹ـ گزینه «۱»	۱۰ـ گزینه «۳»

فصل دوازدهم: الگوریتم‌های پیمایش گراف

۱ـ گزینه «۲»	۲ـ گزینه «۱»	۳ـ گزینه «۱»	۴ـ گزینه «۲»	۵ـ گزینه «۴»
۶ـ گزینه «۳»	۷ـ گزینه «۳»	۸ـ گزینه «۲»	۹ـ گزینه «۲»	۱۰ـ گزینه «۲»
۱۱ـ گزینه «۳»	۱۲ـ گزینه «۱»	۱۳ـ گزینه «۱»	۱۴ـ گزینه «۲»	۱۵ـ گزینه «۳»

فصل سیزدهم: الگوریتم‌های مبتنی بر فاصله

۱ـ گزینه «۳»	۲ـ گزینه «۳»	۳ـ گزینه «۳»	۴ـ گزینه «۴»	۵ـ گزینه «۳»
۶ـ گزینه «۱»	۷ـ گزینه «۱»	۸ـ گزینه «۲»	۹ـ گزینه «۱»	۱۰ـ گزینه «۳»
۱۱ـ گزینه «۱»	۱۲ـ گزینه «۴»	۱۳ـ گزینه «۲»	۱۴ـ گزینه «۲»	۱۵ ـ گزینه «۴»

فصل چهاردهم: مرتب‌سازی‌های مقایسه‌ای

۱ـ گزینه «۴»	۲ـ گزینه «۳»	۳ـ گزینه «۲»	۴ـ گزینه «۲»	۵ـ گزینه «۴»
۶ـ گزینه «۱»	۷ـ گزینه «۱»	۸ـ گزینه «۳»	۹ـ گزینه «۲»	۱۰ـ گزینه «۱»
۱۱ـ گزینه «۳»	۱۲ـ گزینه «۴»	۱۳ـ گزینه «۲»	۱۴ـ گزینه «۴»	۱۵ـ گزینه «۲»
۱۶ـ گزینه «۲»	۱۷ـ گزینه «۳»	۱۸ـ گزینه «۱»	۱۹ـ گزینه «۱»	۲۰ـ گزینه «۴»

فصل پانزدهم: مرتب‌سازی‌های غیرمقایسه‌ای

۱ـ گزینه «۱»	۲ـ گزینه «۲»	۳ـ گزینه «۴»	۴ـ گزینه «۳»	۵ـ گزینه «۲»

فصل شانزدهم: مسأله انتخاب

۱ـ گزینه «۱»	۲ـ گزینه «۳»	۳ـ گزینه «۲»	۴ـ گزینه «۴»	۵ـ گزینه «۳»