سوالات تستی زوج و فرد

سوال ۱ وقتی $T(n) \in o(f(n))$ باشد آنگاه....

۱) t یک کران بالا برای f است.

۲) f یک کران بالا برای T است.

۲) f یک کران بالا برای f است.

۴) نمی توان اظهار نظر کرد.

حل سوال ۱ تستی : گزینه ی ۴ نماد O (بیگ او) یعنی حداکثر به اندازه رشد آن تابع رشد می کند. یعنی در این سوال تابع T حداکثر به اندازه F یا کمتر ازآن $\leftarrow$ F کران بالایی برای T است.

سوال۲) در قطعه برنامه زیر تعداد دفعالت تکرار دستور شماره ۳ کدام است؟

۱) $for(k = O; k < +n - 1\,; k + +$

۲) $for\ (i = 1\,; i < +n - k; i + +$

(3 $a[i][i + k] = k;$

۱) $\frac{n^2}{2}$

۲) $\frac{n(n+1)}{2}$

۳) $\frac{n(n-1)}{2}$

۴) n^2

حل سوال تستی ۲:

تغییرات k	تغییرات i
0	$1,2,3, \dots, n$
1	$1,2,3, \dots, n-1$
2	$1,2,3, \dots, n-2$
.	.
.	.
.	.
$n-1$	1

در نتیجه تعداد اجرای دستور $a[i][i + k] = k$ برابر است با

$$1 + 2 + 3 + 4 + \cdots + n = \frac{n(n + 1)}{2}$$

سوال ۳)کدام گزینه الگوریتم بازگشتی ها را کامل می کند ؟

$$
\begin{array}{l}
void\ honoi\ (int\ n, peg\ A, peg\ B, Peg\ C \\
if(n == 1) \\
move\ topdisk\ on\ A\ to\ C \\
else\ \{ \\
move\ topdisk\ on\ A\ to\ C \\
hanoi\ (n-1, B, A, C \\
\}
\end{array}
$$

۱) $Hanoi\ (n-1, C, A, B)$

۲) $Hanoi\ (n-1, A, B, C)$

۳) $Hanoi\ (n-1, A, C, B)$

۴) $Hanoi\ (n-1, A, B, C)$

پاسخ تستی ۳) اگر n دیسک داشته باشیم و ورودی های محورهای A و B و C باشند.الگوریتم برج هانوی به صورت زیر می باشد

$$
\begin{array}{l}
void\ Hanoi\ line\ n, peg\ A, peg\ B, peg\ C; \\
\{ \\
\quad if\ (n == i) \\
\qquad move\ topDisk\ on\ A\ to\ C; \\
\quad else \\
\qquad \{ \\
\qquad\ Hanoi\ \ (n-1, A, C, B) \\
\qquad\ Move\ top\ Disk\ on\ A\ to\ C; \\
\qquad\ Hanoi\ \ (n-1, B, A, C); \\
\qquad \}
\end{array}
$$

که در بین گزین ها، گزینه و ۳ عبارت کامل کننده الگوریتم است.

سوال ۴) پیچیدگی زمانی تابع بازگشتی زیر کدام است؟

$$
\begin{aligned}
&int\ test(int\ m, int\ n)\\
&\{\\
&\quad if\ (n == 1)\\
&\qquad return\ (m)\\
&\quad else\\
&\qquad return\ \big(m \times test\ (m, n-1)\big);\\
&\}
\end{aligned}
$$

۱) $O(mn)$ ۲) $O(m+n)$ ۳) O(logn)   ۴) $O(n)$

پاسخ ۴ تستی) مثلاً به ازای $n = 4$ تابع به صورت زیر است

$$m * \underbrace{Test(m,3)}$$

$$\downarrow$$

$$m * \underbrace{test(m,2)}$$

$$\downarrow$$

$$m * \underbrace{test\ (m,1)}_{m}$$

و به این ترتیب

خواهیم دید که با اضافه کردن هر واحد به n یکبار انجام تابع اضافه می شود. حداکثر به اندازه n خواهد بود $\leftarrow O(n)$ گزینه ۴

سوال ۵ جواب رابطه بازگشتی زیر چیست؟

۱) $\theta(n\ \log_2 n)$ ۲) $\theta(\log_2 n)$ ۳) $\theta(n)$ ۴) $\theta(\log(\log n))$

پاسخ سوال ۵ تستی)

$$T_1 = 1$$
$$T_{(n)} = 2t\left(\frac{n}{b}\right) + n$$

با توجه به قضیه اصلی شماره ۲

$$\left.\begin{array}{l} T_{(n)} = a\,T\left(\frac{n}{b}\right) + Cn^k \\ T_{(1)} = C \end{array}\right\} \qquad \begin{cases} T_n = \theta\left(n^{\log\frac{a}{b}}\right) & a > b^k \quad \text{اگر} \\ T_n = \theta(n^{k\,log\frac{n}{2}}) & a = b^k \quad \text{اگر} \\ T_n = \theta(n^k) & a < b^k \quad \text{اگر} \end{cases}$$

که در این سوال $a = b^1 \leftarrow k = 1$ و $c = 1 \leftarrow$ $T(n) = \ \theta\left(n^1 \log\frac{n}{1}\right) = \theta(n\,log\,\frac{n}{2})$

گزینه۱

سوال ۶) اگر در محدوده $|2^{k-1}, 2^k|$ باشد آنگاه الگوریتم $Binary\ Seacrt$ حداکثر چند مقایسه برای یک جستجوی ناموفق انجام می دهد؟

۱) فقط k مقایسه ۲) فقط $k-1$ مقایسه ۳) k یا $k-1$ مقایسه ۴) k یا $k+1$ مقایسه

پاسخ سوال ۶ تستی : اگر عدد مورد جستجو در بازه 2^k تا 2^{k-1} باشد، الگوریتم جستجوی باینری برای جستجوی موفق k مقایسه و برای جستجوی ناموفق k یا $k-1$ مقایسه انجام می دهد بستگی به تعداد زوج یا فرد داده ها دارد. گزینه ی ۳

سوال ۷)بدترین حالت زمانی الگوریتم $binsearch$ (جستجوی دودوئی) برای جستجو موفق کدام است؟

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

پاسخ سوال ۷) اگر عدد مورد جستجو در محدوده $[2^{k-1}, 2^k]$ باشد، آنگاه الگوریتم دودویی برای جستجوی موفق k مقایسه و برای جستجوی ناموفق k یا $k-1$ مقایسه انجام می دهد.

سپس جستجوی موفق در بدترین حالت مرتبه زمانی $O(fog\, n)$ و جستجوی ناموفق $\theta(\log n)$ می باشد.

گزینه ی ۳

سوال ۸) پیچیدگی زمانی الگوریتم $Quick\ Sort$ وقتی که داده ها از قبل مرتب شده باشد کدام است؟

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

پاسخ سوال ۸) الگوریتم مرتب سازی سریع بدترین شرایط زمانی رخ می دهد که در مجموعه داده ها هیچ دو یا چند مساوی وجود نداشته باشد و در هر بار فراخوانی $pratition$ یک زیر مجموعه حاصل ، تهی و زیر مجموعه اگر شامل کلیه داده به استثنای عنصر محوری باشد. که این در حالتی است که داده ها از قبل مرتب شده باشد. تابع در بدترین حالت به شکل زیر است.

$$T_{(n)} = T_0 \quad + \quad T(n-1) \quad + \quad n-1$$

زمان لازم برای تقسیم بندی به صورت دوزیر است.	زمان لازم برای مرتب سازی زیر است طرف راست	زمان لازم برای مرتب سازی است طرف چپ

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

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

مجموع جملات فوق

$T(n) = \frac{n(n-1)}{2} \rightarrow T(n) \in O(n^2)$ گزینه ی ۳

سوال ۹) در ضرب ماتریس ها به روش استراسن برای محاسبه هر Cij چند عملگر ضرب یا تفریق استفاده می شود؟

۱)عمگر ضرب و ۹ عملگر جمع یا تفریق ۲) ۱۷ عملگر ضرب و ۸ عملگر جمع یا تفریق

۳)۱۸ عملگر ضرب و ۷ عملگر جمع یا تفریق ۴) عملگر ضرب و ۱۸ عملگر جمع یا تفریق

پاسخ سوال ۹) در روش استراس ابتدا هفت ماتریس به نام های S, R, Q, P و برای ضرب دو ماتریس $n \times n$ V, U, T طبق دستورات زیر ایجاد می شود.

$$P = (A_{11} + A_{22})(B_{11} + B_{22})$$
$$Q = (A_{21} + A_{22})B_{11}$$
$$R = A_{11}(B_{12} + B_{22})$$
$$T = A_{22}(B_{12} + B_{11})$$
$$U = (A_{21} + A_{11})(B_{11} + B_{12})$$
$$V = (A_{12} + A_{22})(B_{21} + B_{22})$$

و C_{ij} های آن طبق دستور زیر به دست می آیند.

$$\mathrm{C} = \mathrm{P} + \mathrm{S} - \mathrm{T} + \mathrm{V}$$
$$\mathrm{C} = \mathrm{P} + \mathrm{T}$$
$$\mathrm{C} = \mathrm{Q} + \mathrm{S}$$
$$\mathrm{C} = \mathrm{P} + \mathrm{R} - \mathrm{Q} + \mathrm{U}$$

و میدانیم که مثلاً $A - B$ یعنی A با قرینه ی B جمع شده، در نتیجه با توجه به عملیات حاصل می بینیم که V عمل ضرب و 18 عمل جمع انجام شده. گزینه ی ۴

سوال ۱۰) در مسئله خرد کردن پول هدف پس دادن باقی مانده پول مشتری با حداقل تعداد سکه ها است در صورتی که بخواهیم باقی مانده پول مشتری را که برابر 32 ریال است. با داشتن سکه های موجود در مجموعه بپردازیم (A) ، راه حل روش حریصانه برای این مسئله کدام زیر مجموعه از سکه ها است ؟

$$A = \{25,20,12,10,5,3,1,1\}$$

۱)$(20,12)$ ۲) $\{25,5,1,1\}$ ۳) $\{20,10,1,1\}$ ۴) $\{12,10,5,3,1,1\}$

پاسخ سوال ۱۰ تستی) در الگوریتم حریصانه مسائله خرد کردن پول ابتدا سراغ بزرگترین سکه می رویم بدون توجه به انتخاب های بعدی و الگوریتم حریصانه در خرد کردن پول بدین صورت الزاماً جواب بهینه را نمی دهد الگوریتم به شکل زیر می باشد.

$$
\begin{array}{l}
SET\ Greedy-Applying\ (C) \\
\quad \{ \\
\qquad C=\{\,25,20,12,10,5,3,1,1\} \\
\qquad S=[0] \\
while\ (!\,soluTion(s)\&\&\ c!=0) \\
\qquad \{ \\
\qquad\quad x=SelecT\ (c); \\
\qquad\quad c=c-\{x\}; \\
\qquad\quad if\ (feasible\ (S,x) \\
\qquad\qquad S=S+\{x\}; \\
\qquad \} \\
\qquad if\left(soloTion\ (s)\right) \\
\qquad\quad return\ S; \\
\quad else \\
\qquad\quad return\ (0); \\
\}
\end{array}
$$

x	$Feasible\ (s,x)$	S	$Solution$	$return$
25	(0,25) yes	25	No	0
20	(25,20) No	25	No	0
12	(25,12) No	25	No	0
10	(25,10) No	25	No	0
5	(25,5) yes	30	No	0
3	(30,3) No	30	No	0
1	(30,1) yes	31	No	0
1	(31,1) yes	32	Yes	32

سکه هایی که $Feasible$ برابر Yes شده انتخاب می شوند $\rightarrow$ گزینه ی ۲ صحیح است $\rightarrow 25+5+1+1$

اما در حالت بهینه دو سکه 20 و 12 نمایشی است اما با الگوریتم حریصانه جواب باالست.

سوال ۱۱) به طور کلی آیا جواب الگوریتم حریصانه بهینه است ؟

۱)همیشه بله　　　　۲)همیشه خیر

۳) در حالت کلی نمی توان به این سوال جواب داد　　　　۴) به ورودی برنامه بستگی دارد

پاسخ سوال ۱۱) به طور کلی نمی توان گفت که آیا الگوریتم حریصانه برای یک مسائله پاسخ بهینه را می دهد یا نه اما اگر دو خاصیت ۱ – انتخاب حریصانه و۲- داشتن بهینه زیر ساختاری در روش حل موجود باشد به جواب بهینه می رسیم . گزینه ی ۳ صحیح است.

سوال ۱۲) فرض کنید برای $n = 7$ کارها مهلت و ارزش مربوط به کارها را به صورت زیر است و ارزش یک کار در صورتی حاصل می شود که آن کار مهلت خود اهدا گردد با استفاده از الگوریتم حریصانه ترتیب بهینه برای کارها با ارزش کل ماکزیمم کدام است؟

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

۱) $\{1,2,3\}$　　　　۲)$\{2,4,1\}$　　　　۳) $\{2,4,7,1\}$　　　　۴) $\{4,2,1,5\}$

پاسخ سوال ۱۲) تعدادی از حالت ممکن انجام ۳ کار به صورت زیر است.

مجموعه ارزشی	سوم	دوم	اول
$60 + 20 + 15 = 95$	کار پنجم	کار چهارم	کار اول
$50 + 60 + 15 = 125$	کار پنجم	کار اول	کار دوم
$10 + 20 + 60 = 90$	کار اول	کار چهارم	کار ششم
$20 + 5 + 60 = 85$	کار اول	کار هفتم	کار چهارم

اما با توجه به اینکه فقط کار های اول و پنجم می تواند در نوبت سوم انجام شوند حالت بهینه زیر خواهد بود

کار اول ۶۰ ، کار چهارم ۲۰ ، کار دوم ۵۰ $= 50 + 20 + 60 = 130$

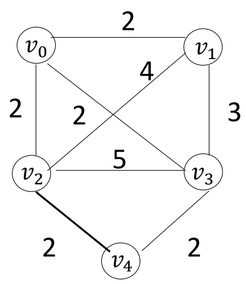
پس ترتیب انجام کار ها در گزینه ۲ صحیح است.

سوال ۱۳)زمان اجرای بهترین الگوریتم برای مساله (خرد کردن مبلغ x ریال با استفاده از n سکه کدام است؟

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

پاسخ سوال ۱۳) الگوریتم بهینه برای مساله خرد کردن پول از درخت بازگشتی استفاده می کند که با توجه به اینکه تعداد زیر مجموعه های فضای حالت n باشد و عمق n هم لگاریتم آن باشد، در نتیجه از مرتبه زمانی تتای $nlogn$ خواهد بود جواب گزینه ی۳ است

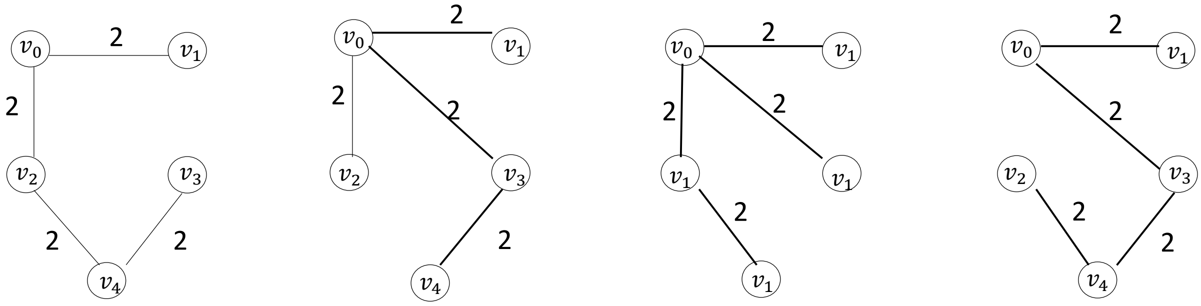
سوال ۱۴) گراف زیر دارای چند پوشای کمینه ای است؟



۱) ۲ ۲) ۳ ۳) ۴ ۴) ۵

پاسخ سوال ۱۴)

طبق درختای پوشاکینه وزن کمینه آن 8 می باشد



در گزینه ۳ صحیح می باشد

سوال ۱۵) تعداد حالت های ممکن برای ضرب ماتریس ها کدام است ؟

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

۳) $\sum_{i=1}^{n-1} T(i-n)$ ۴) $\sum_{i-1}^{n-1} T(i)$

پاسخ سوال ۱۵)

$inT\ minmult(int\ , inT\ m\ [\ ][n+1])$
$\{$
$\quad int\ i, j\ , k;$
$\quad for\ (i = 1\ ; i \leq n\ ; i++)$
$\qquad m[i][j] = 0\ ;$
$for\ (k = 1\ ; k < n\ ; k++$
$\quad for\ (\ i = 1\ ; i \leq n-k\ ; i++$
$\{$
$\quad j = i + L$

$$m[i][j] = min\ (m[i][p] + m\ [\ p+1\ ][\ i\ ] + r[i-1] \times 2\ [k] \times r[i-1] \times r[p] \times r[i]);$$
$\quad i \leq p \leq j$
$\qquad \}$
$\qquad return\ m\ [n][i]$
$\}$

دستورات اجرا برای هر مقدار p را عمل اصلی در نظر می گیریم که در اینجا مقایسه ای است که برای m صورت می گیرد که 3 حلقه تو در تو است . با اندیس های p, j, z چون $j = i + k$ به ازای مقادیر $k\ , i$ تعداد گره از حلقه p عبارت است از $j - 1 - i + 1 = i + k - i - 1 + 1 = k$

به ازای مقادیر معلوم k تعداد گذر ها از حلقه for با اندیس i برابر است با $n - k$ چون k از یک تا $n -$ 1 تغییر می کند و مقدار دفعاتی که عمل اصلی انجام می شود

گزینه ۲ صحیح است

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

سوال ۱۶) تعداد اعمال جمع برای الگوریتم ضریب دو جمله ای $\binom{m}{k}. k \le n$ با استفاده از برنامه نویسی پویا، وقتی که $k = 4$ و $n = k$ است، برابر کدام است؟

پاسخ سوال ۱۶) تعداد اعمال جمع از فرمول زیر محاسبه می شود

$$T(n,k) = K\left(\frac{2n-k-1}{2}\right)$$
$$T(8,4) = 4\left(\frac{16-4-1}{2}\right) = 4 \times \frac{11}{2} = 22$$

گزینه ی ۱صحیح است

سوال ۱۷) تعداد کل گذرها در محاسبه ضریب دو جمله ای زیر با استفاده از برنامه نویسی پویا کدام است؟

۱) $\theta(k \log\ n)$ ۲) $\theta(n \log\ k)$

۳) $\theta(\mathrm{nk})$ ۴) $\theta\ (kn^2)$

پاسخ سوال ۱۷) به ازای k و n مقدار گذرهای انجام شده از حلقه j در الگوریتم ضریب دو جمله ای را محاسبه می کنیم ضریب در جمله $\binom{n}{k}$

```
void bin (inT n , int K)
{
   int i, j;
   int B[n][k];
   for (i = 0 ; j ≤ n ; i + +)
           for(i = 0; i ≤ min(j, k); i + +)
            if (j == 0 | | j == i)
               B [i][j] = 1;
else
              B[i][j] = B [ i − 1][j − 1] + B [i − 1][i]
          return B[n][k]
}
```

جدول زیر تعداد گذرها به ازای بعضی مقادیر i است.

i	۰	۱	۲	...	k	$k+1$	$...n$
تعداد گذرها	۱	۲	۳	...	$k+1$	$k+1$	$...k+1$

سپس تعداد کل گذرها = $1+2+3+4+\cdots+k+(k+1)+(k+1)\ldots+(k+1)$

سپس داریم

$$\frac{k(k+1)}{2}+(n-k+1)(k+1)=\left(\frac{2k-k+2}{2}\right)(k+1)\in\ \theta(nk)$$

<u>گزینه ی ۳ صحیح است</u>

سوال ۱۸) پیچیدگی حافظه و زمانی الگوریتم فروشنده دوره گرد با استفاده از برنامه نویسی پویا در هر حالت به ترتیب از راست به چپ کدام است؟

۱) $\theta\ (n2^n),\theta\ (2^n)$ ۲) $\theta\ (2^n),\theta\ (n2^n)$

۳) $\theta\ (n^2 2^n),\theta\ (n2^n)$ ۴) $\theta\ (n2^n),\theta\ (n^2 2^n)$

پاسخ سوال ۱۸) در الگوریتم برنامه نویسی پویا برای مساله فروشنده دوره گرد سه حلقه for وجود دارد به طوری که حلقه اول و آخر در مقایسه با حلقه دوم زمان زیادی ندارد و بنابراین دستورات اجرا شده برای v_j را عمل اصلی در نظر می گیریم و n تعداد رئوس گراف می باشد، به ازای هر مجموعه A حاوی k راس باید $n-k-1$ راس در نظر بگیریم و به ازای هر راس عمل اصلی k مرتبه اجرا می شود. چون تعداد زیر مجموعه های A از $V-\{v_i\}$ حاوی k راس برابر است با $\binom{n-1}{k}$

مقدار کل دفعاتی که عمل اصلی انجام می شود برابر است با $T(n)=\ \sum_{k=1}^{n=2}(n-1-k)\binom{n-1}{k}$

و می دانیم رابطه ی زیر برقرار است.

$(n-1-k)\binom{n-1}{k}=n-1\binom{n-2}{k}$ ، با قرار دادن این رابطه در رابطه ی v_i خواهیم داشت $T(n)=(n-1)\sum_{k=1}^{n=2}(n-k)k\binom{n-2}{k}$ پس با استفاده از قضیه ی $\sum_{k=1}^{n}k\binom{n}{k}=n2^{n-1}$

نتیجه می گیریم $T(n)=(n-1)(n-2)2^{n-3}\in\theta\ (n^2 2^n)$ و حافظه مورد نیاز آن هم در رابطه $M(n)=2\times n2^{n-1}=n^2 2^n$ به دست می آید <u>پس جواب گزینه ی ۳ است</u>

سوال ۱۹) تعداد درخت های جستجوی دودویی با n عنصر ئ به عمق $n-1$ برابر است؟

۱) n^2 ۲) 2^n ۳) 2^{n-3} ۴) 2^{n-1}

پاسخ سوال ۱۹) درخت های ما با عمق $n-1$ زیر مجموعه ای از کل حالات مورد نظر است یعنی هر گره فقط یک فرزند دارد. در این حالت گره فرزند می تواند سمت چپ یا راست پدر خود باشد یعنی برای قرار دهی هر گره بجز ریشه ۲ حالت انتخاب وجود دارد بنابراین تعداد کل انتخاب ها 2^{n-1} است. یعنی تعداد درختان جستجوی دودویی متفاوت با عمق $n-1$ برابر با 2^{n-1} ← گزینه ی ۴ جواب است

سوال ۲۰) اگر فرمول بهینه بازگشتی زیر برای مسائله پرانتز گذاری ضرب n ماتریس با استفاده از برنامه نویسی پویا به کار رفته باشد؟ به جای علامت ؟ کدام گزینه قرار می گیرد؟

$$m[i,j] = \min\{m[i,k] + m[?,j] + pi - 1pk\ pj - if\,(i<j)if\ (i=j)$$

۱) $k+1$ ۲) k ۳) $k-1$ ۴) $k/2$

پاسخ سوال ۲۰) الگوریتم حداقل تعداد ضرب ها که همان پرانتز گذاری بهینه است در صورت زیر می باشد.

```
inT minmult(int , inT m [ ][n + 1])
{
   int i,j ,L;
   for (i = 1 ; i ≤ n ; i + +)
       m[i][j] = 0 ;
for (l = 1 ; L < n ; L + +
     for ( i = 1 ; i ≤ n − L ; i + +
{
     j = i + L

m[i][j] = mi: n (m[i][k] + m [ k + 1 ][ i ] + r[i − 1] ∗ 2 [k] ∗ v[j])
      i ≤ k < j
          }
          return m [n][i]
}
```

در نتیجه گزینه ی ۱ صحیح می باشد.

سوال ۲۱) تعداد گره ها در درخت فضای حالت برای مسائله رنگ آمیزی گراف با n راس و m رنگ به روش عقبگرد کدام است؟

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

پاسخ سوال ۲۱) الگوریتم زیر مساله رنگ آمیزی گراف را به شیوه عقب گرد حل می کند.

```
void m − coloring (index i)
  {
inT color;
if (pomising (i))
   if(i == n)
  cout ≪ Vcolor[i]through Vcolor [n];
else
    for (color = 1; color  ≤ m; color + +)
    {
    Vcolor  [i + 1] = color;
      m − coloring (i + 1);
    }
  }
bool promising (index i)
{
  index j ;
  bool switch;
swith = true;
j = 1;
while (j < i && switch
 {
  if (w[i][j]&& Vcolor [i} == Vcolor [j]
  switch = flase;
   j + +
  }
  return switch;
}
```

در الگوریتم رنگ آمیزی گراف تابع در سطح بالا به صورت $m-coloring\ (\theta)$ فراخوانی می شود، مقدار گره ها در درخت فضای حالت این الگوریتم برابر با :

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

مانند سایر الگوریتم های عقبگرد دیگر، این الگوریتم نیز می تواند برای یک نمونه بزرگ ویژه بازدهی بالایی داشته باشد.

می توان برای حالت های خاص $m = 2$ (۲رنگ) الگوریتمی نوشت که پیچیدگی زمانی آن در بدترین حالت هم نسبت به n نمایی نباشد اما برای $m \geq 3$ تا کنون الگوریتمی ایجاد نشده که نمایی نباشد گزینه ی ۱ صحیح است.

سوال ۲۲) مرتبه زمانی الگوریتم n و زیر در تکنیک عقبگرد کدام است؟

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

پاسخ سوال۲۲) طبق الگوریتم عقب گرد زیر

$$
\begin{array}{l}
Void\ queens\ (k,n) \\
\{ \\
\ \ int\ i\ ; \\
\ \ for\ (i = 1\ ; i \leftarrow n\ ; i++) \\
\ \ \ \ if\big(pomising\ (k,i)\big) \\
\ \ \ \ \{ \\
\ \ \ \ \ \ x[k] = i; \\
\ \ \ \ \ \ if(k == n) \\
\ \ \ \ \ \ \ \ pint\ (x); \\
else \\
\ \ \ \ \ \ \ \ qeens(k+1,n); \\
\ \ \ \ \} \\
\}
\end{array}
$$

به عنوان مثال برای ۸ وزیر به صورت $qeens(1{,}8)$ فراخوانی می شود این الگوریتم همه جواب ها را تولید می کند برای تحلیل الگوریتم باید قرار گره های چک شده را به عنوان تابعی از n یعنی تعداد وزیرها تعیین کنیم.

حد بالای تعداد گره های درخت فضای حالت عبارت است از

۱- گره در سطح صفر یک، n^2 گره در سطح ۲ و.... n^n گره در سطح 18

$$\rightarrow 1 + n + n^2 + \cdots + n^n = \frac{n^{n+1} - 1}{n - 1}$$

الگوریتم بالا از چک کردن بسیاری از این گره ها جلوگیری می کند مثلاً برای 8 و زیر اولین و زیر در هر یک از 8 ستون می تواند قرار گیرد و دومی را حداکثر در 7 ستون می توان قرار داد و الی آخر

بنابراین در درخت فضای حالت در سطح صفر تنها یک گره، در سطح 1 هشت گره، و در سطح 2 ، 8×7 و... در سطح 8 م و !8 گره وجود خواهد داشت. لذا در حالت کلی

$$1 + 8 + (8 \times 7) + \cdots + 8! = \sum_{i=0}^{8}(\prod_{j=0}^{j}(8 - i) = 69281$$

با تقسیم این نتیجه برای n و زیر حداکثر تعداد گره های موجود در فضای حالت و یاحداکثر تعداد گره های امید بخش عبارت است از

$$1 + n + n(n-1) + n(n-1)(n-2) + \cdots + n!$$

در نتیجه از مرتبه زمانی $\theta(n!)$ خواهد بود. گزینه ۳

سوال ۲۳) الگوی جستجو در درخت به روش بازگشت به عقب و روش انشعاب و تحدید چه تفاوتی با هم دارد؟

۱)در روش بازگشت به عقب جستجوی عرضی است و در انشعاب و تحدید جستجوی عمقی است.

۲) در روش بازگشت به عقب جستجوی عمقی است و در روش انشعاب و تحدید جستجوی عرضی است

۳) در روش جستجوی عرضی است.

۴) در هر دو روش جستجوی عمقی است.

پاسخ سوال ۲۳) عقبگرد حالت اصلاح شده جستجوی عمقی یک درخت است. در روش عقبگرد مجموعه تصمیمات به صورت یک درخت نشان داده می شود به نام درخت تصمیم $(decision\ tree)$ برای گرفتن یک تصمیم درست یا نادرست از یک سطح به سطح دیگر می رویم و تا برگ ها ادامه می دهیم. در این روش هیچ کدام از تصمیمات گرفته شده قطعی نیست مگر اینکه به یک جواب برسیم.

روش انشعاب و تحدید یا پیمایش عرضی است که زیر درخت های هر گره یکی یکی تولید می شود و تا بررسی کامل هر زیر درخت دیگری ایجاد نمی شود گزینه ی ۲ صحیح است.

سوال ۲۴) به ترتیب از سمت راست به چپ کدام مسائله جزو کلاس p و کلاس np است ؟

۱) حاصل ضرب اعداد بزرگ، زنجیره ضرب ماتریس ها
۲) رنگ آمیزی گراف، مساله کوله پشتی
۳) زنجیره ضرب ماتریس ها، رنگ آمیزی گراف
۴) جستجوی دودوئی، حاصلضرب اعداد بزرگ

جواب سوال ۲۴) مسائل P مسائلی هستند که به کمک الگوریتم هایی از مرتبه چندجمله ای قابل حل هستند. و مسائل NP هم توسط الگوریتم های چند جمله ای غیرقطعی قابل حل هستند که در بین عبارت های داخل گزینه ها رنگ آمیزی گراف مربوط به np و زنجیره ضرب ماتریس ها مربوط به کلاس P می باشد ← گزینه ی ۳ صحیح است.

۲۵) مسائلی که الگوریتم کارا برای آنها ابداع نشده ولی غیرممکن بودن آن نیز به اثبات نرسیده است چه مسائلی هستند؟

۱) NP ۲) NP کامل ۳) P ۴) همه موارد صحیح است.

جواب سوال ۲۵) مسائلی که الگوریتم کارا (چند جمله ای) برای آنها ابداع شده ولی غیرممکن بودن آن نیز هنوز اثبات نشده را مسائل NP کامل می گویند.

مسائل NP مجموعه مسائل تصمیم گیری است که توسط الگوریتم های غیر قطعی با زمان چند جمله ای قابل حل هستند.

مسائل P هم مجموعه تمام مسائل تصمیم گیری هستند که توسط الگوریتم های زمانی چند جمله ای قابل حل هستند.

سوالات تشریحی ۱) رابطه ی بازگشتی زیر را حل کنید:

$$T(1) = 1$$
$$(T(n) = 2T(\sqrt{n}) + logn$$

جواب سوال تشریحی ۱) روش غیر متغییر

$$n = 2^m \rightarrow m = log\,{}^{m}_{2}\,, \sqrt{n} = \sqrt{2^m} = 2^{\frac{m}{2}}$$

$$T(n) = T(2^m) = 2T\left(2^{\frac{m}{2}}\right) + \log{}^{2^m}_{2} = 2T\left(2^{\frac{m}{2}}\right) + m$$

فرض کنیم $T(2^m) = S(m) \leftarrow$

$$S(m) = 2S\left(-\right) + m$$

$$\rightarrow S(m) = \theta(m \log \quad) \rightarrow T(n) = Sm = \theta(\log{}^{n}_{2} \times log_2^{(log_2^n)})$$

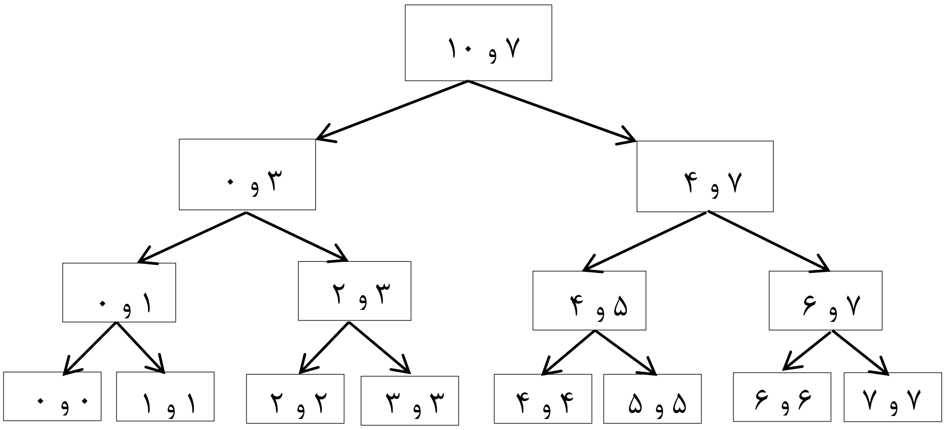
سوال ۲ تشریحی) الگوریتم مرتب سازی ادغای را بر روی لیست زیر، به صو.رت درخت فراوانی $Mergesorth$ اجرا کنید.

پیچیدگی زمانی این الگوریتم در بدترین حالت چیست؟

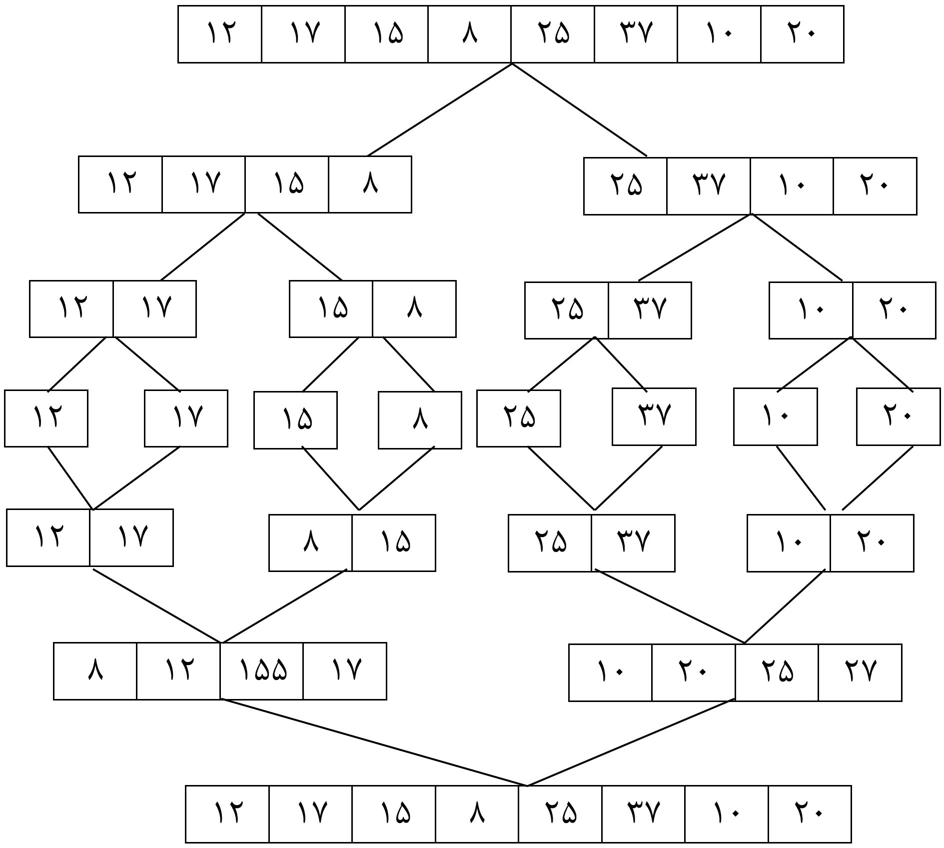
12,17,15,8,25,37,10,20

پاسخ سوال ۲ تشریحی)

آرایه ها به دو زیر است از اندیس 3…. و 7…. 4 تقسیم می شوند پس آن زیر لیست ها هر کدام به دو زیر لیست تقسیم شده تا آخر که در برگ درخت فراخوانی داده های تکی قرار گیرند پس برگ ها را در صورت نیاز جابجا کرده تا مرتب سازی شکل گیرد و در هر مرحله از بازگشت زیر لیست ها را ادغام و مرتب می کنیم



پس داده های 12,17,15,8,25,37,10,20 را طبق شکل صفحه ی بعد مرتب می کنیم.



اعمال اصلی در الگوریتم مرتب سازی ادغامی مقایسه و انتساب است تابع زمانی آن به صورت

$$T(n)=\begin{cases} a & n=1 \\ 2T\left(\frac{n}{2}\right)+c_n & n>1 \end{cases}$$

می باشد که در آن $T\left(\frac{n}{2}\right)$ زمان بازگشت و حل بوده است و

$\theta(n)$ زمان مورد نیاز برای ادغام زیر لیست هاست ← زمان الگوریتم ادغام به شکل زیر است.

$$T(n) = \begin{cases} a & n = 1 \\ 2T\left(\frac{n}{2}\right) + c_n & n > 1 \end{cases}$$

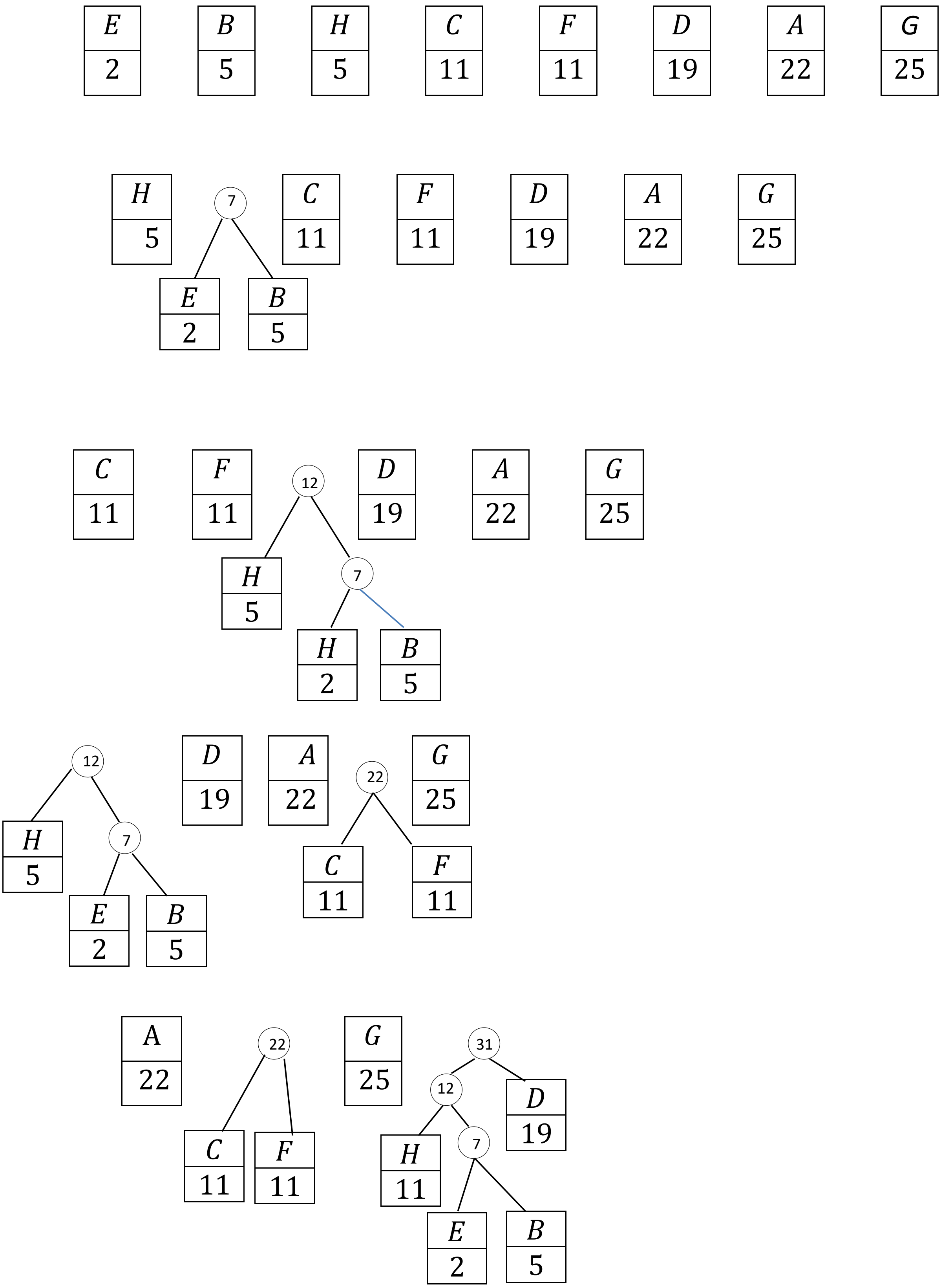
که با تکرار و جایگزینی می توان نوشت.

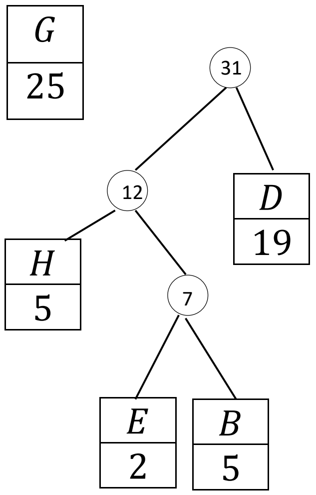
$$\begin{aligned} T(n) &= 2\left(2T\left(\frac{n}{4}\right) + c\frac{n}{2}\right) + Cn \\ &= 4T\left(\frac{n}{4}\right) + 2c \\ &= 4\left(2T\left(\frac{n}{8}\right) + C\frac{n}{4}\right) + 2cn \\ &= \cdots \\ &= 2^k T(1) + kcn \\ &\rightarrow n = 2^k \rightarrow k = \log n \rightarrow T(n) = an + cn\log n \\ &\rightarrow T(n) \in \theta\,(n\log n) \end{aligned}$$

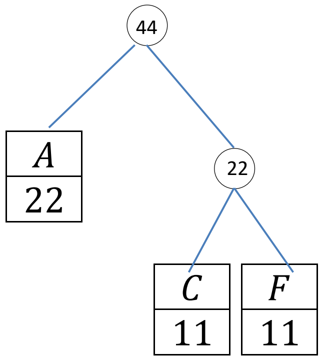
سوال ۳ تشریحی) برای فراوانی زیر درخت هافمن را رسم کرده و کد هر کاراکتر را مشخص کنید

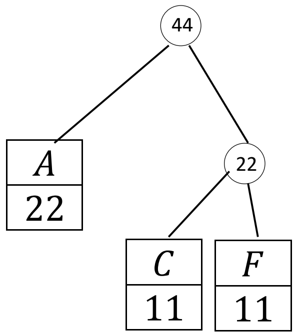
عنصر اطلاعاتی	A	B	C	D	E	F	G	H
وزن	۲۲	۵	۱۱	۱۹	۲	۱۱	۲۵	۵

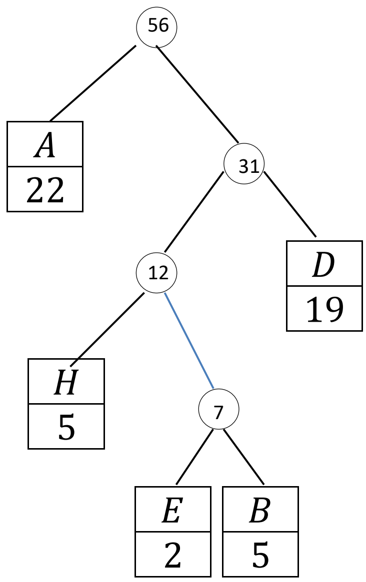
پاسخ سوال ۳ تشریحی عناصر را بر حسب وزن مرتب می کنیم و ابتدا دو عنصر با وزن کمتر را به یک گره والد حاوی مجموع آنها وصل کرده ، سپس ئالد را با سایر عناصر مرتب می کنیم و این روند را تکرار می کنیم تا درخت هافمن حاصل شود

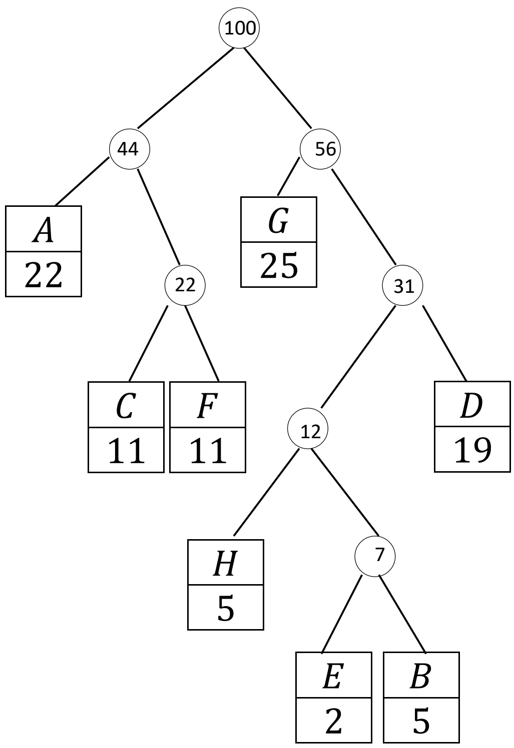

E 2
B 5
H 5
C 11
F 11
D 19
A 22
G 25
7
12
22
31


G
25
31
12
D
19
H
5
7
E
2
B
5


44
A
22
22
C
11
F
11


44
A
22
22
C
11
F
11


56
A
22
31
12
D
19
H
5
7
E
2
B
5


100
44
56
A
22
G
25
22
31
C
11
F
11
12
D
19
H
5
7
E
2
B
5

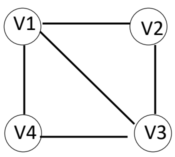
$A = 0\ \ 0$	$D = 111$	$G = 10$
$B = 11\ 0\ \ 11$	$E = 11\ 0\ 10$	$\mathrm{H} = 11\ 00$
$C = 0\ \ \ 1\ \ \ 0$	$F = 0\ 11$	

سوال ۴ تشریحی) مسائله رنگ آمیزی گراف را در نظر بگیرید که در آن هدف رنگ آمیزی گره های گراف $G(V,E)$ با استفاده از m رنگ است. به طوری که هیچ دو گره مجاوری همرنگ نباشند.

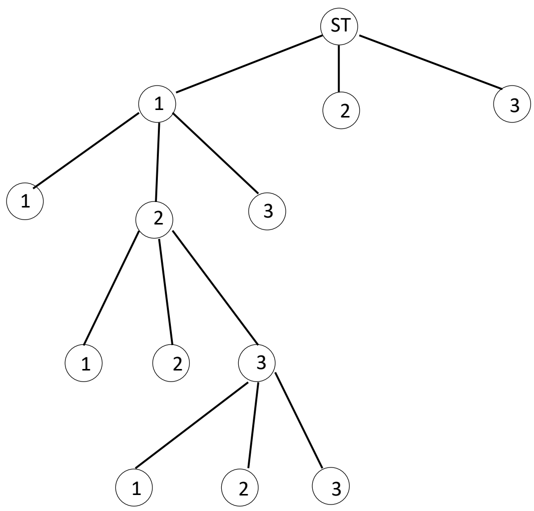
الف) با استفاده از روش عقبگرد مساله را تحلیل نموده و الگوریتم کاملی را برای حل این مساله بنویسید (تابع امید بخش نیز نوشته شود) ؟

ب) مرتبه زمانی الگوریتم برای حل این مساله در بهترین حالت چگونه است؟

پاسخ سوال ۴ تشریحی) درخت فضای حالت برای رنگ آمیزی با m رنگ درختی است که در آن هر رنگ ممکن برای راس V_1 در سطح ۱ 2 امتحان می شود ولی آخر تا اینکه هر رنگ ممکن برای راس V_2 در سطح 2 امتحان می شود ولی آخر تا اینکه هر رنگ ممکن برای راس V_n در سطح n امتحان می شود.

مثلا برای رنگ آمیزی گره های زیر می باشد. 

با ۳ رنگ درخت فضای حالت به صورت زیراست.



```
Void m color (index i)
    int   color;
    if   (promi sing (i))
    if ( i == n )
            cout ≪ Vcolor [i]throuth Vcolor[n];
else
    for (color = 1; color  ≤ m ; color + +)
        {
          Vcolor [i + 1] = color;
          Mcolor (i + 1);
                  }
         }
boo! promising (index i)
{
  index i;
  bool Falg;
  j = 1;
while (i < i &&  Flag)
{
  if (w[i][j]   &&   Vcolor[i] == Vcolor[ij]
         Flag = False;
    j + +;
}
return Flag;
}
```

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

$$1 + m + m^2 + \cdots + m^n = \frac{m^{n+1} - 1}{m - 1} = T(u) = O(m^n)$$

می بینیم که پیچیدگی نمایی است و برای m های بزرگتر مساوی 3 تا کنون الگوریتمی مطرح شده که مرتبه زمانی آن در بدترین حالت بهتر از نمایی باشد.

سوال ۵ تشریحی) یک جواب بهینه برای مساله کوله پشتی صفر و یک زیر در شرایط زیر (با یکی از روش های تکنیک عقبگرد یا روش انشعاب و تحدید) بیابید

شماره کالا	۱	۲	۳	۴
ارزش	۲۰	۴۰	۳۰	۱۰
وزن	۱۲	۲۰	۸	۲

ظرفیت کوله پشتی برابر ۲۰ کیلوگرم می باشد (دقت نمایید مسئله کوله پشتی صفر و مدنظری باشد.

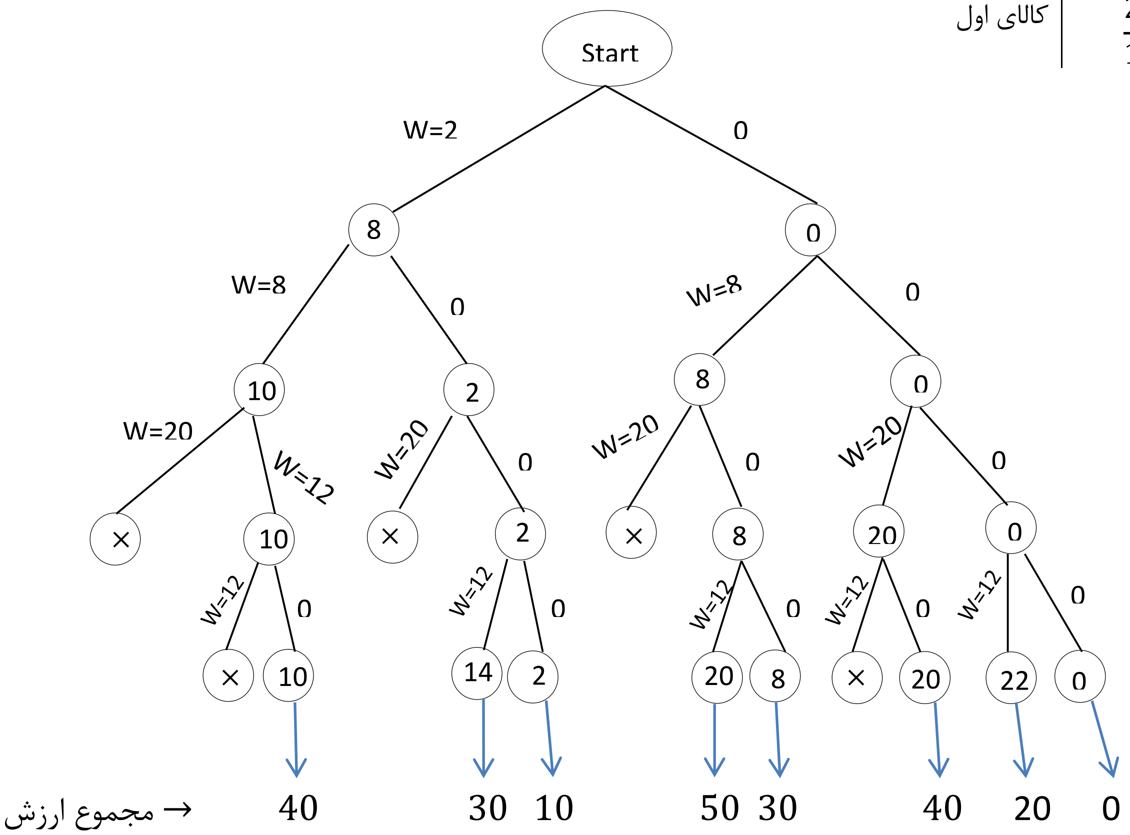
پاسخ سوال ۵ تشریحی) الگوریتم عقبگرد را مسائله کوله پشتی صفر و یک به صورت زیر می باشد.

$$Void\ \ cheeknode\ (nodeV)$$
$$\{$$
$$node\ u;$$
$$if\ (value\ (V) is\ better\ than\ best)$$
$$best = value\ (v);$$
$$if\ \left(pro, ising\ (V)\right)$$
$$for\ (each\ chikd\ u\ of\ v)$$
$$checkmode\ (u);$$
$$\}$$

کالا	۱	۲	۳	۴
ارزش	۲۰	۴۰	۳۰	۱۰
وزن	۱۲	۲۰	۸	۲

مرتب کردن قطعات بر حسب $\frac{pi}{wi}$

	$\frac{pi}{wi}$
کالای چهارم	$\frac{10}{2} = 5$
کالای سوم	$\frac{30}{8} = 3/75$
کالای دوم	$\frac{40}{20} = 2$
کالای اول	$\frac{20}{12}$



جواب بهینه $50 = 20 + 30 =$ کالای 1 + کالای 3

قابل قبول $20 \leq 20 = 12 + 8$ وزن

\end