نام و نام خانوادگی :

محمد حسین توکلی

شماره دانشجویی :

۹۶۳۸۷۴۷۹۲

درس :

طراحی الگوریتم ها

موضوع :

پاسخ سوالات فرد تابستان ۹۸ + سوالات زوج نیمسال دوم ۹۳-۹۴ | ردیف ۱۸

تاریخ :

شهریور ۱۳۹۹

تابستان ۹۸ – سوالات فرد

سوالات تستی:

۱- تعداد گره ها در درخت فضای حالت برای الگوریتم عقبگرد برای مساله رنگ آمیزی m کدام است؟ (m تعداد رنگ ها و n تعداد رئوس گراف می باشد.)

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

پاسخ: بر اساس صفحه ۲۶۹ کتاب، تعداد گره ها برای این الگوریتم برابر است با: $1 + m + m^2 + \cdots + m^n = \sum_{k=0}^{n} m^k = \frac{m^{n+1}-1}{m-1}$. لذا گزینه چهارم صحیح است.

۳- پیچیدگی زمانی حاصل ضرب دو ماتریس n x n کدام است؟

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

پاسخ: بر اساس صفحه ۱۰۷ کتاب، برای ضرب دو ماتریس nxn به n ضرب نیاز داریم و از آنجا که ماتریس حاصلضرب دارای n^2 عنصر است، با ضرب تعداد n عملیات ضرب در n^2 عنصر، زمان لازم برای بدست آوردن ماتریس حاصلضرب $\theta(n^3)$ خواهد بود.

۵- دو مرحله روش حدس و استقرا کدام است؟

۱. حدس جواب، به کارگیری استقرا ریاضی برای یافتن متغیر ها
۲. حدس جواب، به کارگیری استقرا ریاضی برای یافتن ثابت ها
۳. یافتن قطعی جواب، به کارگیری استقرا ریاضی برای یافتن متغیر ها
۴. یافتن قطعی جواب، به کارگیری استقرا ریاضی برای یافتن ثابت ها

پاسخ: بر اساس صفحه ۵۳ کتاب، روش حدس و استقرا برای حل رابطه بازگشتی دارای دو مرحله ی حدس جواب و بکارگیری استقرا ریاضی برای یافتن ثابتها و نشان دادن صحت حدس اولیه می باشد. همچنین با توجه به نام روش، مشخص است که در روش مذکور دنبال یافتن جواب قطعی نیستیم و نیز متغیرها همانگونه که از نامشان پیداست متغیر بوده و قرار نیست از طریق استقرا آنها را بیابیم. لذا گزینه های یک و سه و چهار اشتباه بوده و تنها گزینه دوم صحیح است.

۷- یکی از روش های خوب برای حل یا حدس روابط بازگشتی از طریق تکرار، استفاده از کدام روش است؟

۱. روش مرتب سازی ادغامی
۲. روش مرتب سازی سریع
۳. روش درخت بازگشت
۴. روش بهینه سازی

پاسخ: بر اساس صفحه ۶۲ کتاب، در روش درخت بازگشت، مقدار اولیه عبارت غیر بازگشتی در ریشه درخت قرار می گیرد. در سطح بعدی درخت به تعداد جملات بازگشتی گره ایجاد می شود و تا زمانی که n به مقدار ثابت نرسیده تشکیل سطح درخت تکرار می شود. لذا یکی از روش های خوب حل یا حدس روابط بازگشتی از طریق تکرار، روش درخت بازگشت می باشد.

۹- زمان جستجوی موفق در بدترین حالت در درخت تصمیم دودوئی کدام است؟

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

پاسخ: بر اساس صفحه ۵۳ کتاب، اگر X در محدوده $[2^{k-1},2^k]$ باشد آنگاه الگوریتم BinSrch حداکثر k مقایسه عنصر برای یک جستجوی موفق انجام می دهد. و چون $k-1 \leq \text{Log } n < k$ ، لذا زمان یک جستجوی موفق در بدترین حالت O(Log n) خواهد بود.

۱۱- پیچیدگی زمانی الگوریتم مرتب سازی سریع در بدترین حالت و حالت متوسط به ترتیب از راست به چپ کدام است؟

۱. $\theta(n\,Ln\,n)\,,\theta(n^2)$ ۳. $\theta(n^2)\,,\theta(n\,Ln\,n)$

۲. $\theta(n^2)\,,\theta(n^2)$ ۴. $\theta(n\,Ln\,n)\,,\theta(n\,Ln\,n)$

پاسخ: بر اساس صفحه ۱۰۴ کتاب، در بدترین حالت داریم:

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

$$T(n)=T(n-1)+(n-1)=T(n-2)+(n-2)+(n-1)=\cdots=\frac{n(n-1)}{2}\Longrightarrow \boldsymbol{T(n)\in\theta(n^2)}$$

و بر اساس صفحه ۱۰۶ کتاب، در حالت متوسط داریم:

$$T(n)=\sum_{i=1}^{k}S_i\,P_i=\left(\frac{1}{n}\right)\sum_{P=1}^{n}T(P-1)+T(n-P)+n-1 \text{ و } a_n=\frac{T(n)}{n+1}\Longrightarrow a_n=\begin{cases}0 & if\ n<1\\ a_{n-1}+\dfrac{2(n-1)}{n(n+1)} & if\ n\geq 1\end{cases}$$

$$a_n=a_{n-1}+2\left(\frac{2n-1}{n}-\frac{2n}{n+1}\right)=\cdots\leq\sum_{i=1}^{k}\frac{1}{k}=2\mathrm{Ln\ n}\Longrightarrow T(n)\leq 2(n+1)Ln\,n\Longrightarrow \boldsymbol{T(n)\in\theta(n\,Ln\,n)}$$

لذا بر اساس مقادیر به دست آمده، گزینه اول صحیح است.

۱۳- در کدام الگوریتم زیر، برای یافتن کلیه کوتاهترین مسیرها از مبدا واحد به مقصدهای متفاوت به کار می رود و همچنین طول یک مسیر را برابر مجموع وزن یال های آن مسیر در نظر می گیرد؟

۱. الگوریتم پریم ۲. الگوریتم دیکسترا ۳. الگوریتم کروسکال ۴. الگوریتم فلوید

پاسخ: بر اساس صفحه ۱۵۷ کتاب، الگوریتم دیکسترا کلیه کوتاهترین مسیرها از منبع واحد به مقصدهای متفاوت را محاسبه می کند. طول یک مسیر را برابر مجموع وزن یال های آن مسیر در نظر می گیرد. همچنین رئوس شروع کننده روی مسیر را به عنوان راس منبع یا همان source و آخرین راس را به عنوان راس مقصد یا همان Destenation می شناسد.

۱۵- کدام ویژگی در خصوص مسائلی که به روش برنامه نویسی پویا حل می شود، به درستی بیان شده است؟

۱. در همه الگوریتم های برنامه نویسی پویا، مساله بهینه سازی موضوعی کلیدی است.

۲. مسائل را از بالاترین سطح به طرف پایین ترین سطح حل می کند.

۳. در هر سطح، بعضی از مسائل آن سطح حل می گردند و بقیه به سطح بعد منتقل می شود.

۴. برای حل هر مساله سطح L می توانیم از کلیه مسائل سطوح پایین تر که لازم باشد، استفاده کنیم.

پاسخ: بر اساس صفحه ۱۹۶ کتاب، بر خلاف روش تقسیم و حل، که برای حل هر مساله سطح L تنها از مسائل سطح L-1 استفاده می کند، در روش برنامه نویسی پویا برای حل هر مساله سطح L می توانیم از کلیه مسائل سطوح پایین تر که لازم باشد استفاده کنیم. همچنین بر اساس همین صفحه، در اغلب الگوریتم های برنامه نویسی پویا مساله بهینه سازی کلیدی است، نه در همه ی موارد. و نیز در الگوریتم برنامه نویسی پویا، مسائل را از پایین ترین سطح بطرف بالاترین سطح حل می کنیم و نه برعکس. و نیز در هر سطح کلیه مسائل موجود آن سطح حل می گردند و نتایج نگهداری می شوند، نه بعضی مسائل. لذا گزینه های یک تا سه اشتباه بوده و گزینه چهارم صحیح است

۱۷- پیچیدگی زمانی مساله فروشنده دوره گرد، با استفاده از روش برنامه نویسی پویا کدام است؟

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

پاسخ: بر اساس صفحه ۲۲۳ کتاب، داریم:

$$\sum_{k=1}^{n} k\binom{n}{k} = n\,2^{n-1} \text{ و } T(n) = \sum_{k=1}^{n-2} (n-1-k)k\binom{n-1}{k} = (n-1)\sum_{k=1}^{n-2} k\binom{n-2}{k}$$

$$T(n) = (n-1)(n-2)2^{n-3} \Longrightarrow \boldsymbol{T(n) \in \theta(n^2 2^n)}$$

لذا بر اساس مقادیر محاسبه شده، گزینه سوم صحیح است.

۱۹- الگوریتم های عقبگرد برای حل مسائلی از قبیل کوله پشتی صفر و یک، کدام پیچیدگی زمانی را دارد؟

۱. خطی ۲. نمایی ۳. بدتر از نمایی ۴. بهتر از نمایی

پاسخ: بر اساس صفحه ۲۵۲ و نیز ۲۸۲ کتاب، الگوریتم عقبگرد برای مسائلی از قبیل کوله پشتی صفر و یک در بدترین حالت باز هم نمایی هستند و تنها با کاهش حالت ها زمان اجرا را کاهش می دهند. لذا گزینه دوم صحیح است.

۲۱- فضای مساله ای که با استفاده از روش انشعاب و تحدید حل می شود باید چگونه نمایش داده شود؟

۱. باید با یک درخت قابل نمایش باشد.
۲. باید با یک پشته قابل نمایش باشد.
۳. باید با یک گراف قابل نمایش باشد.
۴. باید با یک لیست پیوندی قابل نمایش باشد.

پاسخ: بر اساس صفحه ۲۸۷ کتاب، فضای حالت مساله ای که قرار است با استفاده از روش انشعاب و تحدید حل گردد، باید با یک گراف قابل نمایش باشد. لذا گزینه سوم صحیح است.

۲۳- مسائلی که الگوریتم کارا (چند جمله ای) برای آنها ابداع نشده است ولی غیر ممکن بودن آن نیز هنوز به اثبات نرسیده، کدام مسائل هستند؟

۱. P ۲. Np ۳. Np-hard ۴. Np کامل

پاسخ: بر اساس صفحه ۳۱۱ کتاب، مسائلی که الگوریتم کارا برایشان ابداع نشده ولی غیر ممکن بودنشان نیز هنوز به اثبات نرسیده را مسائل Np کامل گویند. لذا با توجه به اینکه P توسط الگوریتم های قطعی با مرتبه زمانی چند جمله ای قابل حل هستند و Np نیز توسط الگوریتم های غیر قطعی با مرتبه زمانی چند جمله ای می توانند قابل حل باشند که ممکن است الگوریتم چند جمله ای برای حل مساله داشته باشیم و یا نداشته باشیم و با توجه به اینکه Np-hardنیز حداقل به سختی سختترین مسائل در ان پی است و حتی ممکن است برخی مسائل آن در Np نباشد. لذا تنها گزینه چهارم صحیح است.

۲۵- الگوریتم رام نشدنی کدام است؟

۱. الگوریتم هایی با مرتبه زمانی n، n^2 و n^3 را مسائل رام نشدنی می نامند.
۲. مسائلی که نوشتن یک الگوریتم کارآمد برای آنها غیر ممکن است مسائل رام نشدنی می گویند.
۳. الگوریتم هایی که مرتبه زمانی آنها چند جمله ای باشد را مسائل رام نشدنی می نامند.
۴. الگوریتم هایی که مرتبه زمانی آنها logn و nlogn باشد را مسائل رام نشدنی می گویند.

پاسخ: بر اساس صفحه ۳۰۶ کتاب، مسائلی که نوشتن یک الگوریتم کارآمد برای آنها غیر ممکن است را مسائل رام نشدنی Intractable می گویند که الگوریتم زمانی آنها غیر چند جمله ای است. و چون مسائلی که الگوریتم های زمانی آنها چند جمله ای باشد حلشان ساده بوده و رام نشدنی به حساب نمی آیند، لذا گزینه های یک و سه و چهار رام نشدنی نمی باشند و تنها گزینه دوم صحیح است.

سوالات تشریحی:

۱- الگوریتم بازگشتی برای محاسبه فاکتوریل یک عدد نوشته و زمان اجرای الگوریتم را تحلیل کنید.

پاسخ: می دانیم:

$$n! = \begin{cases} 1 & if\ n = 0 \\ n(n-1)! & if\ n > 0 \end{cases}$$

تابع بازگشتی:

```
int fact (int n)
{
    If (n==0)
        return (1);
    else
        return (n* fact(n-1));
}
```

با توجه به تابع بازگشتی فوق الگوریتم محاسبه فاکتوریل را برای n=3 شرح می دهیم:

fact (3) [ ] → fact (2) [A:3] → fact (1) [A:3 , B:2] → fact (0) [A:3 , B:2 , C:1]
return (6) [ ] ← return (2) [A:3] ← return (1) [A:3 , B:2] ← return (1) [A:3 , B:2 , C:1]

قبل از فراخوانی اول پشته خالی بوده

در اولین فراخوانی آدرس مقدار n در پشته ذخیره می شود ، از مقدار n یکی کم شده و تابع مجددا فراخوانی می شود

عملیات بالا تا جایی ادامه می یابد که n=0 شود

در n=0 مقدار یک بازگشت داده می شود و عملیات بازگشت آغاز می شود

به ازای هر مرحله بازگشت یک عمل حذف از بالای پشته انجام می گیرد

تا زمانی که پشته خالی نشده باشد عمل بازگشت ادامه می یابد

با خالی شدن پشته، مقدار محاسبه شده توسط تابع همان مقدار بازگشت شده نهایی خواهد بود

زمان اجرای الگوریتم فاکتوریل:

در حالت کلی، تابع n بار فراخوانی می شود و تعداد گره های درخت بازگشت آن، زمان اجرای الگوریتم خواهد بود. لذا $T(n) \in O(n)$

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

$$T(n) = T(n-1) + C = T(n-2) + 2C = \cdots = T(1) + (n-1)C = T(0) + nC - (n+1)C \Rightarrow \boldsymbol{T(n) \in O(n)}$$

۳- رابطه $T(n) = T\left(\left\lceil \frac{n}{2} \right\rceil\right) + 1$ را با روش حدس و استقرا حل نمایید.

پاسخ:

حدس:

$$T(n) \in O(\text{Log } n) \Rightarrow \exists\, C > 0 :\ T(n) \leq C \text{ Log } n$$

پایه استقرا:

$$n = 2 \Rightarrow T(2) \leq C \log 2 = C \Rightarrow C \geq T(2) > 0\ \checkmark$$

فرض استقرا:

$$\forall\, k < n : T(k) \leq C log_2 k$$

حکم استقرا:

$$\forall\, n : T(n) \leq C log_2 n$$

$$T(n) \leq C \log_2 \left(\left\lceil \frac{n}{2} \right\rceil\right) + 1 \leq C \log n - C + 1 \leq C \log n \Rightarrow \forall\, \boldsymbol{C \geq 1 : T(n) \in O(\log n)}$$

۵- اجزا تشکیل دهنده یک الگوریتم حریصانه را نام برده و شرح دهید.

پاسخ: اجزاء تشکیل دهنده یک الگوریتم حریصانه عبارت اند از:

- مجموعه ای از انتخاب های ممکن برای مولفه های جواب به نام C و مجموعه مولفه های انتخاب شده تا به حال به نام S .
- روالی به نام Select برای انتخاب مولفه های بعدی جواب از مجموعه انتخاب های ممکن.
- روالی به نام Feasible که عنصر انتخاب شده توسط روال Select را جهت قرار گرفتن در مجموعه جواب یا رد آن بررسی می کند.
- روالی به نام Solution برای بررسی اینکه مشخص کند در نهایت جواب حاصل شده است یا خیر.
- یک تابع هدف که هدف بهینه کردن این تابع است.

نیمسال دوم ۹۳-۹۴ – سوالات زوج

سوالات تستی:

۲- در الگوریتم زیر در صورتی که n=m باشد مرتبه اجرایی برابر است با:

```
For i=1 to n Do
    For j:=1 to m do
        For k:=1 to j do
            X:=x+1;
```

۱. $O\left(\frac{m+1}{2}\right)$ ۲. $O(n^2)$ ۳. $O\left(\frac{m(m+1)}{2}\right)$ ۴. $O(n^3)$

پاسخ: بر اساس صفحه ۸ کتاب، سطر اول n+1 بار، سطر دوم (m+1)*n بار، سطر سوم (m+1)*m*n بار و سطر چهارم m*m*n بار با ثابت C1 وC2 و C3 و C4 اجرا می شوند. با در نظر گرفتن ثابت C به عنوان بیشترین مقدار C1 تا C4 ، خواهیم داشت:

$T(n) = (n+1)C1 + (m+1)nC2 + (m+1)mnC3 + m^2nC4$ و $n = m \Longrightarrow T(n) = C(2n + 2n^2 + 2n^3 + 1)$

$\Longrightarrow \boldsymbol{T(n)} \in \mathbf{O(n^3)}$

همچنین می توان گفت با در نظر گرفتن ثابت زمانی C، خواهیم داشت:

$T(n) = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} C = n*m*m*C$ و $n = m \Longrightarrow T(n) = n^3C \Longrightarrow \boldsymbol{T(n)} \in \mathbf{O(n^3)}$

لذا گزینه چهارم صحیح است.

۴- در ضرب سه آرایه A(3,4) , B(4,6) , C(6,2) به ترتیب A*B*C چند عمل ضرب انجام می شود؟

۱. ۲۵ ۲. ۱۰۸ ۳. ۲۵۹۲ ۴. ۳۴۵۶

پاسخ: بر اساس صفحه ۱۹۹ کتاب، ضرب ماتریس ixj در ماتریس jxk ، تعداد i x j x k عملیات ضرب خواهد داشت. با توجه به اینکه در مساله فوق یکبار ماتریس A در ماتریس B ضرب می شود و یکبار حاصلضرب ماتریس A و B در ماتریس C ضرب می گردد، لذا تعداد عملیات ضرب از مجموع تعداد عملیات هر دو ضرب حاصل خواهد شد. لذا گزینه دوم صحیح است

تعداد عملیات ضرب A_{3X4} X B_{4x6} : 3*4*6 = 72

تعداد عملیات ضرب $(A \text{ X } B)_{3x6}$ X C_{6x2} : 3*6*2 = 36

مجموع عملیات ضرب : 72 +36 = 108

۶- مرتبه زمانی رابطه بازگشتی T(n)=9T(n/3)+n برابر است با:

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

پاسخ: بر اساس صفحه ۶۵ کتاب، طبق قضیه اصلی داریم:

$$\begin{cases} a = 9 \\ b = 3 \\ F(n) = n \end{cases} \Longrightarrow n^{\log_b a} = n^2 \Longrightarrow \begin{cases} F(n) \in O\left(n^{\log_b a - \varepsilon}\right) \\ \varepsilon > 0 \end{cases} \Longrightarrow T(n) \in O\left(n^{\log_b a}\right) \Longrightarrow \boldsymbol{T(n)} \in \boldsymbol{O(n^2)}$$

با توجه به قضیه، مرتبه زمانی از رابطه $T(n) \in O\left(n^{\log_b a}\right)$ حاصل خواهد شد. لذا طبق محاسبات فوق گزینه اول صحیح است.

۸- چند مورد از عبارات زیر صحیح می باشد؟

- الگوی جستجو برای روش عقبگرد به صورت جستجو در پهنا می باشد.
- در روش انشعاب و تحدید روش جستجوی درخت به ترتیب عمق می باشد.
- در هر دو روش بازگشت به عقب و انشعاب و تحدید شاخه هایی از درخت هرس می شود.

۱. ۳ ۲. ۲ ۳. ۱ ۴. ۰

پاسخ: بر اساس صفحه ۲۸۸ و ۲۸۹ کتاب، روش عقبگرد به صورت جستجو در عمق می باشد و حال آنکه در روش انشعاب و تحدید جستجو به ترتیب پهنا می باشد. اما در هر دو روش شاخه هایی از درخت هرس می شود و درنتیجه زمان لازم برای اجرای الگوریتم کاهش می یابد. لذا با غلط بودن عبارت اول و دوم صحیح بودن عبارت سوم، تنها یک عبارت صحیح است.

۱۰- کدامیک از مرتبه زمانی های زیر جزو مسائل رام نشدنی نمی باشد؟

۱. 2^n ۲. 3^n ۳. n^4 ۴. $n!$

پاسخ: بر اساس صفحه ۳۰۶ کتاب، هر الگوریتمی که مرتبه زمانی آن غیر چند جمله ای باشد (یعنی نمایی باشد) را مسائل رام نشدنی می نامند. لذا عبارات به توان n و یا عبارات فاکتوریلی n به صورت نمایی بوده و رام نشدنی است و تنها n^4 که چند جمله ای است پاسخ صحیح سوال می باشد.

۱۲- تعداد درخت های جستجو با عمق n-1 برابر است با:

۱. 2^n ۲. 2^{n-1} ۳. 2^{n+1} ۴. 3^{n+1}

پاسخ: بر اساس صفحه ۲۱۴ و ۲۴۴ کتاب، با در نظر گرفتن عمق n-1 ، تعداد n سطح و هر سطح یک گره خواهیم داشت که گره سطح اول یا سطح ریشه یک حالت و در سایر سطوح هر گره دو حالت (راست یا چپ) خواهد داشت. لذا تعداد درخت با عمق n-1 برابر خواهد بود با حاصلضرب تعداد حالت گره ها. $1*2^{n-1}=2^{n-1}$ = 1*2*2*2*...*2 . لذا گزینه دوم صحیح است.

۱۴- الگوریتم تولید کننده کد هافمن ، ..... .

۱. همیشه درخت بهینه تولید می کند.
۲. گاهی اوقات درخت بهینه تولید می کند.
۳. هیچ وقت درخت بهینه تولید نمی کند.
۴. اغلب اوقات درخت بهینه تولید می کند.

پاسخ: بر اساس صفحه ۱۸۳ کتاب، با توجه به اینکه سیستم کد گذاری تعریف شده در الگوریتم تولید کد هافمن از نوع درخت دودویی بوده و همواره بهینه خواهد بود، لذا الگوریتم مذکور همواره درخت بهینه تولید می کند.

۱۶- کدام الگوریتم یالی را (از بین رئوس همسایه) در هر مرحله انتخاب می کند که منجر به حداقل افزایش در مجموع هزینه ها می گردد؟

۱. کروسکال ۲. پریم ۳. سولین ۴. دیکسترا

پاسخ: بر اساس صفحه ۱۴۲ کتاب، الگوریتم پریم با یک گره دلخواه شروع می کند، یالی را انتخاب می کند که منجر به حداقل افزایش در مجموع هزینه هایی گردد که تا به حال درنظر گرفته شده است. و زمانی که کلیه گره ها افزوده شود کار پایان می یابد. لذا مجموع هزینه یال های این درخت کمترین مقدار است.

۱۸- کدام روش پیشنهاد می کند که می توان الگوریتمی نوشت که مرحله به مرحله اجرا شود و در هر زمان یک ورودی را بررسی نماید و بررسی انجام شده در مورد شدنی بودن یا نبودن جواب ها می باشد؟

۱. روش تقسیم و حل ۲. حریصانه ۳. برنامه نویسی پویا ۴. عقبگرد

پاسخ: بر اساس صفحه ۱۳۴ کتاب، روش حریصانه پیشنهاد می کند که می توان الگوریتمی نوشت که مرحله به مرحله اجرا شود و در هر زمان یک ورودی را برسی نماید که بررسی انجام شده در مورد شدنی بودن یا نبودن جواب ها می باشد. اگر نتیجه بررسی منفی باشد، جواب نشدنی بوده و در غیر اینصورت، ورودی بررسی شده توسط روالی به نام select به مجموعه جواب افزوده می شود.

۲۰- بکارگیری روش تقسیم و حل برای کدامیک از مسئله های زیر مناسب نمی باشد؟

۱. سری فیبوناچی
۲. مرتب سازی ادغام
۳. مرتب سازی سریع
۴. ضرب ماتریس ها به روش استراسن

پاسخ: بر اساس صفحه ۱۲۰ کتاب، بکارگیری روش تقسیم و حل برای طراحی الگوریتم مسائلی با اندازه n که اندازه زیر مسئله های آن نیز تقریبا برابر n و یا n/c (c ثابت است) می باشد، مناسب نیست. لذا در مسائلی مانند سری فیبوناچی به دلیل اینکه هر زیر مسئله تقریبا به اندازه مسئله اولیه طول دارد، پیچیدگی زمانی الگوریتم آن از نوع نمایی بوده و مناسب نمی باشد. همچنین بر اساس صفحه ۹۷ کتاب، الگوریتم مرتب سازی ادغامی با پیچیدگی زمانی $\theta(n \log n)$ ، و بر اساس صفحه ۱۰۴ کتاب، الگوریتم مرتب سازی سریع با پیچیدگی زمانی $\theta(n^2)$ ، و بر اساس صفحه ۱۱۱ کتاب، الگوریتم ضرب ماتریس ها به روش استراسن با پیچیدگی زمانی $\theta(n^2)$ ، هر سه در طراحی الگوریتم خود از روش مستقیم و حل استفاده می کند.

۲۲- بدترین حالت الگوریتم Quick sort چه زمانی رخ می دهد؟

۱. داده ها از قبل به صورت صعودی مرتب شده باشند.
۲. داده ها از قبل به صورت نزولی مرتب شده باشند.
۳. داده ها از قبل مرتب شده باشند.
۴. به وضعیت ورودی داده ها بستگی ندارد.

پاسخ: بر اساس صفحه ۱۰۳ کتاب، در الگوریتم QUICK SORT بدترین شرایط زمانی رخ می دهد که در مجموعه داده ها، هیچ دو یا چند مجموعه برابر وجود نداشته باشد و در هر بار فراخوانی پارتیشن، یک زیر مجموعه حاصل، تهی و زیر مجموعه دیگر شامل کلیه داده ها به استثنای عنصر محوری باشد. و این حالت زمانی رخ می دهد که این مجموعه داده ها از قبل مرتب شده باشند. با توجه به اینکه گزینه اول و دوم حالت خاصی از پاسخ بوده و کامل نیستند و گزینه چهارم نیز شرط اولیه را نقض می کند، لذا تنها گزینه سوم صحیح است.

۲۴- زمان یک جستجوی موفق در بدترین حالت در الگوریتم جستجوی دودوئی برابر است با:

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

پاسخ: بر اساس صفحه ۵۳ کتاب، اگر X در محدوده $[2^{k-1},2^k]$ باشد آنگاه الگوریتم BinSrch حداکثر k مقایسه عنصر برای یک جستجوی موفق انجام می دهد. و چون $k-1 \leq \text{Log } n < k$ ، لذا زمان یک جستجوی موفق الگوریتم دودویی در بدترین حالت O(Log n) خواهد بود.

سوالات تشریحی:

۲- فرض کنید لیستی حاوی عناصر 17,20,10,25,11,8,18 باشد. با استفاده از مرتب سازی سریع این لیست را مرتب نمایید.

پاسخ: اولین عنصر را به عنوان عنصر محور انتخاب میکنیم. عناصر کوچکتر را در سمت چپ و عناصر بزرگتر را در سمت راست لیست قرار می دهیم. سپس زیر لیست های ایجاد شده را همانند مرحله قبل مرتب سازی می کنیم.

- در اولین مرحله عدد ۱۷ به عنوان عنصر محور انتخاب می شود. تابع QuickSort با مقادیر ۰ و ۶ فراخوانی می شود. با اجرای تابع پارتیشن، عناصر لیست به ترتیب زیر مرتب می شوند

17 20 10 25 11 08 18
17 10 20 25 11 08 18
17 10 11 25 20 08 18
17 10 11 08 20 25 18
08 10 11 17 20 25 18

- در مرحله بعد زیر لیست سمت چپ عنصر محور اولیه (۱۷) مرتب خواهد شد
- عدد ۸ به عنوان عنصر محور انتخاب می شود. تابع QuickSort با مقادیر ۰ و ۲ فراخوانی می شود. با اجرای تابع پارتیشن، عناصر لیست مرتب می شوند (چون از قبل مرتب بوده نتیجه تغییری نخواهد کرد)

08 10 11 17 20 25 18

- عدد ۱۰ به عنوان عنصر محور انتخاب می شود. تابع QuickSort با مقادیر ۱ و ۲ فراخوانی می شود. با اجرای تابع پارتیشن، عناصر لیست مرتب می شوند (چون از قبل مرتب بوده نتیجه تغییری نخواهد کرد)

08 10 11 17 20 25 18

- عدد ۱۱ به عنوان عنصر محور انتخاب می شود. تابع QuickSort با مقادیر ۲ و ۶ فراخوانی می شود. با اجرای تابع پارتیشن، عناصر لیست مرتب می شوند (چون از قبل مرتب بوده نتیجه تغییری نخواهد کرد)

08 10 11 17 20 25 18

- زیر لیست سمت چپ مرتب شده و در مرحله بعد زیر لیست سمت راست عنصر محور اولیه (۱۷) مرتب خواهد شد.
- عدد ۲۰ به عنوان عنصر محور انتخاب می شود. تابع QuickSort با مقادیر ۱ و ۲ فراخوانی می شود. با اجرای تابع پارتیشن، عناصر لیست به ترتیب زیر مرتب می شوند

08 10 11 17 20 25 18
08 10 11 17 20 18 25
08 10 11 17 18 20 25

- با تغییر اندیس عنصر محور به عدد ۶ (عنصر آخر لیست)، عملیات مرتب سازی با خروجی زیر پایان می یابد.

08 10 11 17 18 20 25

۴- الگوریتم فلوید را نوشته و پیچیدگی زمانی این الگوریتم را بدست آورید.

پاسخ: می دانیم:

$$D^k(i,j) = \min\{D^{k-1}(i,j)\ , D^{k-1}(i,k) + D^{k-1}(k,j)\}$$

```
void Floyd (int n , float W[ ] [n] , float D[ ] [n-1])
{
    int i , j , k ;
    D = W ;
    for (k = 1 ; k ≤ n ; k ++)
        for (i = 0 ; i < n ; i ++)
          for (j = 0 ; j < n ; j ++)
              D[i][j] = min(D[i][j] , D[i][k] + D[k][j]) ;
}
```

W : آرایه دو بعدی گراف جهت دار و وزن دار با n راس که W[i][j] وزن یال بین دو راس i و j می باشد.

D : آرایه دو بعدی که سطرها و ستونهای آن از ۰ تا n-1 اندیس گذاری شده و W[i][j] وزن کوتاهترین مسیر بین راس i و j می باشد.

در این الگوریتم k حالت از ۱ تا n را در نظر میگیریم. برای هر حالت یک سطر i را در نظر میگیریم. به ازای هر سطر، ستون های j متناظر با آن را در نظر میگیریم. برای هر ستون کمترین مقدار آرایه به نسبت حالت انتخاب شده را در D آن سطر و ستون جایگذاری میکنیم. عملیات جایگزین کردن کمترین مقدار $D_{i,j}$ به نسبت k را برای تمام n ستون و n سطر و در تمام n حالت k انجام میدهیم. با پایان یافتن هر سه حلقه تو در تو، مقدار نهایی آرایه D، کوتاهترین مسیر ها را دربرخواهد داشت.

پیچیدگی زمانی : با توجه به سه حلقه تو در تو با عمل اصلی محاسبه مقدار، خواهیم داشت:

$$T(n) = \boldsymbol{n} * \boldsymbol{n} * \boldsymbol{n} = \boldsymbol{n^3} \Rightarrow \boldsymbol{T(n)} \in \boldsymbol{\theta(n^3)}$$