نمونه سوالات فرد ۳_۹۸_۹۷ و زوج ۲_۹۴_۹۳

محدثه روحانی

۹۷۰۱۴۸۰۴۹

تابستان ۹۹

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.** $\frac{n^{m+1}-1}{n-1}$ **۴.** $\frac{m^{n+1}-1}{m-1}$

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

$Fori := 1 to ndo$

$Forj := 1 to mdo$

$Fork := 1 to jdo$

$X := x + 1;$

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

۳. پیچیدگی زمانی حاصل ضرب دو ماتریس $n \times n$ کدام است؟

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

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. 25 **۲.** 108 **۳.** 2592 **۴.** 3456

۵. دو مرحله روش حدس و استقرا کدام است؟

۱. حدس جواب، به کار گیری استقرا ریاضی برای یافتن متغیر ها

۲. حدس جواب، به کار گیری استقرا ریاضی برای یافتن ثابت ها

۳. یافتن قطعی جواب، به کار گیری استقرا ریاضی برای یافتن متغیر ها

۴. یافتن قطعی جواب،به کار گیری استقرا ریاضی برای یافتن ثابت ها

۶. مرتبه زمانی رابطه بازگشتی مقابل برابر است با:

$$T(n) = 9T(n/3) + n$$

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

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

۸. چند مورد از عبارات زیر صحیح می باشد؟

ـ الگوی جستجو برای روش عقبگرد به صورت جستجو در پهنا می باشد.

ـ در روش انشعاب و تحدید روش جستجوی درخت به ترتیب عمق می باشد.

ـ در هر دو روش بازگشت به عقب و انشعاب و تحدید شاخه هایی از درخت هرس می شود.

۱. 3
۲. 2
۳. 1
۴. 0

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. $O(\log n)$

۱۰. کدام یک از مرتبه زمانی های زیر جزو مسائل رام نشدنی نمی باشد؟

۱. 2^n ۲. 3^n ۳. n^4 ۴. $n!$

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

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

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}

۱۳. در کدام الگوریتم زیر ، برای یافتن کلیه کوتاهترین مسیرها از مبدا واحد به مقصد های 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

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

۱. همیشه درخت بهینه تولید می کند. ۲. گاهی اوقات درخت بهینه تولید می کند.

۳. هیچوقت درخت بهینه تولید نمی کند. ۴. اغلب اوقات درخت بهینه تولید می کند.

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.

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

۱۷. پیچیدگی زمان 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. $\theta(n^2)$

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

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

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

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.

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.

۲۳. مسائلی که الگوریتم کارا (چندجمله ای) برای آنها ابداع نشده است ولی غیرممکن بودن آن نیز هنوز به اثبات نرسیده ، کدام مسائل هستند؟

۱. P
۲. Np
۳. $Np-hard$
۴. Np

۲۴. زمان جستجوی موفق در بدترین حالت در درخت تصمیم دودوئی برابر است با :

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

۲۵. الگوریتم رام نشدنی کدام است؟

۱. الگوریتم هایی که با مرتبه زمانی n ، $n2$ و $n3$ را مسائل رام نشدنی می نامند.
۲. مسائلی که نوشتن یک الگوریتم کارآمد برای آنها غیرممکن است مسائل رام نشدنی می گویند.
۳. الگوریتم هایی که مرتبه زمانی آنها چندجمله ای باشد را مسائل رام نشدنی می نامند.
۴. الگوریتم هایی که مرتبه زمانی آنها $\log n$ ، $n \log n$ باشد را مسائل رام نشدنی می گویند.

تشریحی

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

۲. فرض کنید لیستی حاوی عناصر زیر باشد:

۱۷،۲۰،۱۰،۲۵،۱۱،۸،۱۸

با استفاده از مرتب سازی سریع این لیست را مرتب نمائید.

۳. رابطه ذکر شده را با روش حدس و استقرا حل نمایید؟

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

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

۵. اجزای تشکیل دهنده یک الگوریتم حریصانه را نام برده و شرح دهید؟

۱	د
۲	د
۳	د
۴	ب
۵	ب
۶	الف
۷	ج
۸	ج
۹	د
۱۰	ج

۱۱	الف
۱۲	ب
۱۳	ب
۱۴	الف
۱۵	د
۱۶	ب
۱۷	ج
۱۸	ب
۱۹	ب
۲۰	الف

۲۱	ج
۲۲	ج
۲۳	د
۲۴	الف
۲۵	ب
۲۶	
۲۷	
۲۸	
۲۹	
۳۰	

پاسخ نامه تستی

ج۱. تعداد گره ها در درخت فضای حالت برای این الگوریتم برابر است با:

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

ج۲.

$$For i := 1 to do \rightarrow \text{بار} n \rightarrow O(n)$$

$$For j := 1 to do \rightarrow \text{بار} m = n \rightarrow O(n) \rightarrow O(n^2)$$

$$For k := 1 to do \rightarrow \text{بار} j = m = n \rightarrow O(n) \rightarrow O(n^3)$$

$$(x := x + 1) \rightarrow O(n^3)$$

ج۳.

$$for(i = 0; i < n; i++)$$

$$for(j = 0; j < n; j++)$$

$$\{$$

$$c[i][j] = 0;$$

$$for(k = 0; k < n; k++)$$

$$c[i][j] = c[i][j] + A[i][k] \times B[k][j]$$

$$\}$$

بنابراین $O(n^3)$

ج۴.

$$A_{3\times4} \times B_{4\times6} \times C_{6\times2}$$

ابتدا ضرب ها

$$\left.\begin{array}{c} A_{3\times4} \times B_{4\times6} = 3 \times 4 \times 6 = 72 \Rightarrow D_{3\times6} \\ \Rightarrow A_{3\times4} \times D_{4\times2} \\ D_{3\times6} \times C_{6\times2} = 3 \times 6 \times 2 = 36 \Rightarrow E_{3\times2} \end{array}\right\} = 72 + 36 = 108 \qquad (۱)$$

ج۵. حدس جواب، به کار گیری استقرا ریاضی برای یافتن ثابت ها

ج۶. از قضیه اصلی استفاده می نماییم:

$T(n) = aT(\frac{n}{b}) + n^k$ $\qquad$ $T(n) = 9T(\frac{n}{3}) + n \rightarrow a = 9, b = 3, k = 1$

a با b^k مقایسه می شود

اگر $a > b^k$ باشد مرتبه زمانی $n^{\log_b^a}$ خواهد بود.

اگر $a < b^k$ باشد مرتبه زمانی n^k خواهد بود.

اگر $a = b^k$ باشد مرتبه زمانی $n^k \log n$ خواهد بود.

$a > b^k \rightarrow T(n) = O(n^{\log_b^a}) = O(n^{\log_3^9}) = O(n^2)$

ج۷. یکی از روش های خوب برای حل یا حدس روش های بازگشتی روش درخت بازگشت است

ج۸. روش انشعاب و تحدید ، بسیار مشابه روش عقبگرد است که از درخت فضای حالت برای حل مسائل استفاده می کند، ۲ تفاوت بین این ۲ روش وجود دارد. در یک روش انشعاب و تحدید:

۱ .محدودیتی برای استفاده از روش خاصی برای پیمایش درخت فضای حالت وجود ندارد.

۲ .فقط برای مسائل بهینه سازی به کار می رود.

فضای حالت مسئله به روش انشعاب و تحدید باید با یک گراف قابل نمایش باشد.

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

۱. DFS ـ جستجوی عمقی ـ مربوط به روش عقبگرد می باشد

۲. BFS ـ جستجوی ردیفی ـ مربوط به روش انشعاب و تحدید می باشد.

در روش انتخاب و تحدید برخلاف روش عقبگرد، امکان تغییر ترتیب بررسی گره ها وجود دارد.

مسئله ای که به روش عقبگرد حل می شود، می تواند بیش از یک جواب داشته باشد و هیچ جوابی بر جواب دیگر امتیازی ندارد، ولی در روش انشعاب و تحدید مهم یافتن جواب بهینه است.

در یک الگوریتم انشعاب و تحدید، برای هر گره،عددی (کرانه ای) برای تعیین امیدبخش بودن آن محاسبه می شود . عدد مزبور، کرانه برای جوابی است که در صورت گسترش گره مربوطه می توان به آن رسید. اگر این کرانه، بهتر از مقدار بهترین جواب حاصل تاکنون نباشد، در این صورت ، گره غیرامیدبخش و در غیر این صورت، امید بخش است.

ج۹. فرض کنید n عنصر $A_N, \ldots, A_2, A_1$ داده شده است و می خواهیم در یک درخت جستجوی دودوئی اضافه شوند . برای n عنصر تعداد $n!$ جایگشت وجود دارد. هر یک از چنین جایگشتی باعث به وجود امدن درخت مربوط به خود می شود. می توان نشان داد که عمق میانگین $n!$ درخت تقریبا برابر با $c \log_2 n$ است که در آن $c = 1.4$ می باشد . بنابراین زمان اجرای میانگین جتسجو یک عنصر در درخت دودوئی T با n عنصر متناسب با $\log_2 n$ است یعنی $f(n) \in O(\log_2 n)$

ج۱۰. زیرا n^4 چند جمله ای است.

مسائلی که نتوان برای آنها الگوریتمی با مرتبه زمانی چند جمله ای پیدا کرد مسائل رام نشدنی نامیده می شود. الگوریتم هایی با مرتبه زمانی $n!, 3^n, 2^n$ یا هر الگوریتمی که مرتبه زمانی آن غیر چند جمله ای باشد را مسائل رام نشدنی می نامند. مسئله تعیین کلیه مدارهای هامیلتونی جزو مسائلی هستند که رام نشدنی بودن آنها اثبات شده است. در این مسائل با توجه به عبارت محاسبه تعداد مدارها می توان دریافت که پیچیدگی زمانی این مسائل !n می باشد.

ج۱۱. بدترین حالت زمانی است که آرایه به صورت صعودی یا نزولی مرتب شده باشد. برای آرایه صعودی داریم:

$$T(n) = \underbrace{T(0)}_{\text{زمان لازم برای مرتب سازی زیرآرایه سمت چپ چون هیچ عنصری در سمت چپ آرایه نیست}} + \underbrace{T(n-1)}_{\text{زمان لازم برای مرتب سازی زیرآرایه سمت راست}} + \underbrace{n-1}_{\text{زمان لازم برای افراز}}$$

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

درنتیجه تعداد مقایسه ها در بدترین حالت : $\frac{n(n-1)}{2} \leftarrow \theta(n^2) \leftarrow$ مرتبه ی زمانی حالت متوسط :

$$T(n) = \underbrace{T(\frac{n}{2})}_{\text{زمان لازم برای مرتب سازی زیرآرایه سمت چپ زیرآرایه سمت راست}} + \underbrace{T(\frac{n}{2})}_{\text{زمان لازم برای مرتب سازی}} + \underbrace{cn}_{\text{زمان لازم برای افراز}}$$

$$T(n) = \begin{cases} 0 & if n < 1 \\ 2T(\frac{n}{2}) + cn & if n \geq 1 \end{cases}$$

در نتیجه درحالت متوسط : $\leftarrow \theta(n \log n) \leftarrow$ مرتبه ی زمانی

ج۱۲. با n گره ، عمق $n-1$ یعنی در هر سطح فقط یک گره داشته باشیم یا فرزند چپ باشد یا فرزند راست.

با n گره می توان ابتدا یک گره را به عنوان ریشه در نظر گرفت سپس در عمق ۱ برای گره بعدی دو حالت داریم (فرزند چپ یا راست).در عمق ۲ هم برای گره بعدی دو حالت داریم و... لذا تعداد کل حالت ها برابر است با:

$\overset{(2\times 2\times \cdots \times 2)}{\text{بار } n-1} = 2^{n-1}$

چون گره ریشه فقط یک حالت دارد در نظر نمیگیریم لذا $n-1$ گره دیگر می توانند دو حالت داشته باشند.

ج۱۳. الگوریتم دیکسترا برای یافتن کلیه کوتاهترین مسیرها از مبدا واحد به مقصد های متفاوت به کار می رود. این الگوریتم همچنین طول یک مسیر را برابر مجموع وزن یال های آن مسیر در نظر می گیرد.

ج۱۴. در کد گذاری هافمن حروف با تکرار بیشتر در گره های نزدیک ریشه قرار می گیرد لذا با طول کمتر خواهد بود و حروف با تکرار کمتر در فاصله بیشتری از ریشه قرار می گیرد و طول کد آنها بیشتر خواهد بود.

ج۱۵. در روش برنامه نویسی پویا کلیه مسائل موجود که در آن سطح حل می گردند نتایجش نگهداری می شود که اگر در سطوح بعد از آنها استفاده شد نیاز به محاسبه مجدد نباشد لذا برنامه نویسی پویا ممکن است نسبت به روش تقسیم و حل در زمان کمتری حل شود.

ج۱۶. مرتبه زمانی الگوریتم پریم $O(n^2)$ و الگوریتم کروسکال $O(eloge)$ می باشد پس زمان اجرای آنها روی گراف های یکسان نمی تواند مساوی باشد. n) راس و e یال)

در گراف متراکم (تعداد یال کمتر) چون تعداد رئوس بیشتر از یال ها است الگوریتم کروسکال سریعتر است و در گراف کامل چون تعداد یال ها بیشتر است، پریم سریعتر عمل می کند. ولی خروجی هردو الگوریتم درخت پوشا با حداقل هزینه و یکسان می باشد.

با توجه به سوال مشخص می شود که تعداد یال های گراف کم می باشد لذا از الگوریتم کروسکال استفاده می شود.

ج۱۷. حل مسئله فروشنده دوره گرد به روش پویا دارای مرتبه زمانی $n^2 2^n$ می باشد و میزان حافظه مورد نیاز $n2^n$ است.

ج۱۸. روش حریصانه برای حل الگوریتم های زیر کاربرد دارد:

الگوریتم بقیه دادن پول

الگوریتم دیکسترا

الگوریتم هافمن

الگوریتم prim

الگوریتم کوله پشتی

الگوریتم های ادغام بهینه

الگوریتم کروسکال

الگوریتم زمان بندی

ج۱۹. نمایی

روش عقبگرد برای حل مسائل زیر کاربرد دارد:

مسئله n وزیر

مسئله رنگ آمیزی گراف ها

مسئله کوله پشتی صفر و یک

مسئله حاصل جمع زیرمجموعه ها

مسئله مدارهای هامیلتونی

ج۲۰. برای مسائلی که با تقسیم مسئله اصلی ، مسئله کوچکتر دوباره به اندازه تقریبا n باشد و مسائلی که به تعداد زیادی زیر مسئله با طول $\frac{n}{C}$ تقسیم می شوند، روش تقسیم و حل مناسب نیست.

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

ج۲۱. روش انشعاب و تحدید ، بسیار مشابه روش عقبگرد است که از درخت فضای حالت برای حل مسائل استفاده می کند، ۲ تفاوت بین این ۲ روش وجود دارد. در یک روش انشعاب و تحدید:

۱.محدودیتی برای استفاده از روش خاصی برای پیمایش درخت فضای حالت وجود ندارد.

۲.فقط برای مسائل بهینه سازی به کار می رود.

فضای حالت مسئله به روش انشعاب و تحدید باید با یک گراف قابل نمایش باشد.

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

۱. DFS ـ جستجوی عمقی ـ مربوط به روش عقبگرد می باشد

۲. BFS ـ جستجوی ردیفی ـ مربوط به روش انشعاب و تحدید می باشد.

ج۲۲. بدترین حالت زمانی است که آرایه به صورت صعودی یا نزولی مرتب شده باشد. برای آرایه صعودی داریم:

$$T(n) = \underbrace{T(0)}_{\text{زمان لازم برای مرتب سازی زیرآرایه سمت چپ چون هیچ عنصری در سمت چپ آرایه نیست}} + \underbrace{T(n-1)}_{\text{زمان لازم برای مرتب سازی زیرآرایه سمت راست}} + \underbrace{n-1}_{\text{زمان لازم برای افراز}}$$

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

درنتیجه تعداد مقایسه ها در بدترین حالت : $\frac{n(n-1)}{2} \leftarrow \theta(n^2) \leftarrow$ مرتبه ی زمانی حالت متوسط :

$$T(n) = \underbrace{T(\frac{n}{2})}_{\text{زمان لازم برای مرتب سازی زیرآرایه سمت چپ}} + \underbrace{T(\frac{n}{2})}_{\text{زمان لازم برای مرتب سازی زیرآرایه سمت راست}} + \underbrace{cn}_{\text{زمان لازم برای افراز}}$$

$$T(n) = \begin{cases} 0 & if n < 1 \\ 2T(\frac{n}{2}) + cn & if n \geq 1 \end{cases}$$

در نتیجه درحالت متوسط : $\leftarrow \theta(n \log n) \leftarrow$ مرتبه ی زمانی

ج۲۳. مسائلی که الگوریتم کارا برای آنها ابداع نشده ولی غیرممکن بودن آن نیز به اثبات نرسیده مسائل NP کامل هستند.

ج۲۴. پیچیدگی الگوریتم جستجوی دودوئی به وسیله تعداد مقایسه های مورد نیاز برای تعیین مکان item در آرایه مشخص می شود. از طرفی میدانیم که، آرایه دارای n عنصر می باشد . با توجه به الگوریتم ملاحظه می شود هر مقایسه در الگوریتم باعث می شود که، اندازه ورودی نصف شود از این رو حداکثر $T(n)$ مقایسه لازم است تا مکان عنصر item پیدا شود، بنابراین تعداد مقایسه ها برابر خواهد بود با:

$$2^{T(n)-1} > n \text{یا} T(n) = [Log_2 n] + 1$$

یعنی زمان اجرا در بدترین حالت برابر $O(Log_2^n)$ می باشد.

ج۲۵. درحالت کلی ، در برخورد با مسائل به ۳ گروه از راه حل ها با مرتبه های زمانی زیر می رسیم:

۱.مسائلی که الگوریتم های زمانی چند جمله ای برای آنها پیدا شده است مثل مرتب سازی ، ضرب ماتریس ها ، ضرب زنجیری ماتریس ها ، جستجو در یک آرایه ، پیمایش گراف ها، درخت پوشای کمینه، کوتاه ترین مسیر بین دو گره، جستجوی دودویی و ... که حل آن ها ساده است.

۲.مسائلی که رام نشدنی بودن آنها ثابت شده است . مثل مسئله مشخص کردن تمام دور های هامیلتونی یک گراف. رام نشدنی بودن تعداد نسبتا کمی از مسائل اثبات شده است.

۳.مسائلی که رام نشدنی بودن آنها اثبات نشده است، ولی هیچ الگوریتم زمانی چند جمله ای هم برای آنها پیدا نشده است. مثل مسئله کوله پشتی صفر و یک، مسئله ی فروشنده ی دوره گرد، مسئله ی حاصل جمع زیر مجموعه ها ، مسئله ی رنگ آمیزی m گراف به ازای $m \geq 3$ و مسئله ی مدار های هامیلتونی

پاسخ نامه تشریحی

ج۰۱

$$
\begin{aligned}
&int fact int(n)\\
&\{\\
&\quad if\ (n == 0)\\
&\qquad return(1);\\
&\quad else\\
&\qquad return(n * fact(n-1));\\
&\}
\end{aligned}
$$

نکته ۱

$$
\begin{cases} T(0) = 1 \\ T(n) = T(n) + c \end{cases} \Rightarrow (aT(n-k)+c)\ ,\ a = 1 \Rightarrow O(n)
$$

ج۰۲

$$
\begin{aligned}
&\overset{17}{(pivot)}\quad \underline{20}\quad 10\quad 25\quad 11\quad \underline{8}\quad 18\\
\Rightarrow\ &\overset{17}{(pivot)}\quad 8\quad 10\quad \underline{25}\quad \underline{11}\quad 20\quad 18\\
\Rightarrow\ &\overset{17}{(pivot)}\quad 8\quad 10\quad \underline{11}\quad 25\quad 20\quad 18\\
\Rightarrow\ &\underbrace{\overset{11}{(pivot)}\quad 8\quad \underline{10}}\quad 17\quad \underbrace{\overset{25}{(pivot)}\quad 20\quad \underline{18}}\\
\Rightarrow\ &\underbrace{\overset{10}{(pivot)}\quad \underline{8}}\quad 11\quad 17\quad \underbrace{\overset{18}{(pivot)}\quad \underline{20}}\quad 25\\
&8\quad 10\quad 11\quad 17\quad 18\quad 20\quad 25
\end{aligned}
$$

ج۳.

$$T(n) = aT(\frac{n}{b}) + f(n)\ ,\ f(n) \notin n^k$$

$$T(n) = \begin{cases} O(n^{\log_b^a}) & if \quad n^{\log_b^a} > O(f(n)) \\ O(O(f(n))) * \log n & if \quad n^{\log_b^a} = O(f(n)) \\ O(f(n)) & if \quad n^{\log_b^a} < O(f(n)) \end{cases}$$

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

$$a = 1, b = 2, f(n) = n\log n$$

$$n^{\log_2^1} < O(n\log n) \Rightarrow T(n) = \theta(n\log n)$$

ج۴. الگوریتم فلوید برای محاسبه کوتاه ترین مسیر از هر راس در یک گراف موزون به رئوس دیگر به کار می رود.

در این روش ماتریس های D^0 تا D^n را به ترتیب به دست می آوریم، جواب مسئله ماتریس D_n خواهد بود و در آن کوتاه ترین مسیرها برای ۲ گره مشخص شده است.

ماتریس D^0 همان ماتریس مجاورت گراف می باشد که در آن:

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

فرمول کلی زیر برای الگوریتم فلوید به کار می رود:

$$D^k(i,j) = \begin{cases} 0 & if \quad i = j \\ \min\left(D^{k-1}(i,k) + D^{k-1}(k,j), D^{k-1}(i,j)\right) & if \quad i \neq j \\ & i \leq k \leq j \end{cases}$$

مرتبه زمانی الگوریتم فلوید$O(n^3)$ می باشد.

ج۵. در روش حریصانه تصمیم گیری به سرعت و با توجه به اطلاعات موجود انجام می گیرد و توجه زیادی به اثرات و عوارض این تصمیم نمی شود به همین خاطر الگوریتم های حاصل صریح و ساده اند.

خصوصیات کلی یک الگوریتم حریصانه عبارتند از:

نتیجه نهایی یک الگوریتم حریصانه مجموعه ای از داده هاست که ممکن است ترتیب آنها نیز اهمیت داشته باشد. این مجموعه از داده ها اکثرا زیر مجموعه داده های ورودی هستند.

مجموعه جواب به صورت مرحله ای بود و در هر مرحله یک مولفه از جواب حاصل می شود.

جواب نهایی باید تابع هدف را بهینه کند.

S مجموعه جواب که در ابتدا تهی است.

Solution بررسی می کند که آیا جواب نهایی حاصل شده است یا نه

select یک عنصر از مجموعه C انتخاب می کند.