

تابستان ۹۸

۱) تعداد گره ها در درخت فضای حالت برای الگوریتم عقبگرد برای مسئله رنگ آمیزی m کدام است؟
(m تعداد رنگ ها و n تعداد رئوس گراف میباشد)

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

پاسخ) گزینه د

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

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

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

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

پاسخ) گزینه د

حافظه مورد نیاز از $o(n^2)$ است. و هر به روز رسانی نیز از $O(n)$ است. پس پیچیدگی زمانی از $o(n^3)$ است.

۵) دو مرحله روش حدس و استقرا کدام است؟

۱. حدس جواب، به کار گیری استقرا ریاضی برای یافتن متغیرها
۲. حدس جواب، به کار گیری استقرا ریاضی برای یافتن ثابت ها
۳. یافتن قطعی جواب، به کار گیری استقرا ریاضی برای یافتن متغیرها
۴. یافتن قطعی جواب، به کار گیری استقرا ریاضی برای یافتن ثابت ها

پاسخ) گزینه ب

۱. حدس جواب

۲. بکار گیری استقرای ریاضی برای یافتن ثابت ها و نشان دادن صحت حدس اولیه.

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

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

پاسخ) گزینه ج

در این روش نحوه جایگذاری یک عبارت بازگشتی و نیز مقدار ثابتی را که در هر سطح از آن عبارت به دست می آید نشان داده میشود. با جمع کردن مقادیر ثابت تمام سطوح جواب به دست می آید.

۹) زمان جستجوی موفق در بدترین حالت در درخت تصمیم دودویی کدام است؟

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

پاسخ) گزینه د

عملیات اصلی بر روی یک درخت جستجوی دودویی به زمانی متناسب با ارتفاع درخت احتیاج دارد. برای یک درخت دودویی کامل با n گره چنین عملیاتی در بدترین حالت در زمان $o(\log n)$ اجرا میشود. بنابراین اگر درخت یک زنجیر خطی با n گره باشد همین عملیات در زمان بدترین حالت $o(n)$ اجرا میشود. امید ریاضی ارتفاع یک درخت جستجوی دودویی که به تصادف ساخته شده است برابر با $o(\log n)$ است. بنابراین عملیات اصلی مجموعه پویا بر روی چنین درختی در حالت میانگین به زمان $o(\log n)$ احتیاج دارد.

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

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

پاسخ) گزینه الف

پیچیدگی زمانی اجرای الگوریتم در بهترین حالت $\theta(n \log n)$ و در بدترین حالت $\theta(n)$ است. با استفاده محاسبات ریاضی می توان نشان داد در حالت متوسط نیز مرتبه اجرا $\theta(n \log n)$ است.

۱۳) در کدام الگوریتم زیر، برای یافتن کلیه کوتاهترین مسیرها از مبدا واحد به مقصد های متفاوت به کار میرود و همچنین طول یک مسیر را برابر مجموع وزن یال های آن مسیر در نظر میگیرد؟

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

پاسخ) گزینه ب

دیکسترا الگوریتمی است که برای پیدا کردن کوتاهترین مسیر بین دو گره در گراف به کار می رود. این گراف، ممکن است نشانگر شبکه جاده ها یا موارد دیگری باشد الگوریتم دایجسترا دارای انواع گوناگونی است. الگوریتم اصلی، کوتاهترین مسیر بین دو گره را پیدا می کند؛ اما نوع متداول تر این الگوریتم، یک گره یکتا را به عنوان گره مبدا (آغازین) در نظر می گیرد و کوتاهترین مسیر از مبدا به دیگر گره ها در گراف را با ساختن درخت کوتاهترین مسیر پیدا می کند.

۱۵) کدام ویژگی در خصوص مسائلی که به روش برنامه نویسی حل میشود، به درستی بیان شده است؟

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

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

۳. در هر سطح، بعضی از مسائل آن سطح حل میگردند و بقیه به سطح بعد منتقل میشود.

۴. برای حل هر مسئله سطح L میتوانیم از کلیه مسائل سطوح پایین تر که لازم باشد، استفاده کنیم.

پاسخ) گزینه د

۱۷) پیچیدگی زمانی مسئله 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) گزینه ج

این الگوریتم به طور مستقیم در مرتبه زمانی $O(n!)$ حل می شود اما اگر به روش برنامه نویسی پویا برای حل آن استفاده کنیم مرتبه زمانی آن $O(n^2 \cdot 2^n)$ خواهد شد که جز مرتبه های نمایی است. باید توجه داشت علی رغم آنکه مرتبه نمایی مذکور زمان بسیار بدی است اما همچنان بسیار بهتر از مرتبه فاکتوریل می باشد. شبه کد الگوریتم فوق به صورت زیر است که در آن تعداد زیر مجموعه های یک مجموعه n عضوی ۲ به توان n می باشد و for اول یک ضرب n را نیز حاصل می شود که به ازای تمام شهرهای غیر مبدأ می باشد و حاصل $n \cdot (2^n)$ را پدیدمی آورد. بنابراین برای جستجوی کمترین مقدار نیاز به یک عملیات خطی از مرتبه n داریم که در زمان فوق نیز ضرب می شود و در نهایت زمان $(n^2 \cdot 2^n)$ را برای این الگوریتم حاصل می کند. $\leq \theta(n^2 2^n)$

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) گزینه ب

۲۱) فضای مسئله ای که با استفاده از روش انشعاب و تحدید حل میشود 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) گزینه ج.

روش انشعاب و تحدید یکی دیگر از روش های پیمایش و جستجوی درختها و گرافهاست. فضای حالت مسئله ای که قرار است به روش انشعاب و تحدید حل گردد باید با یک گراف قابل نمایش باشد.

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) گزینه د

در تئوری پیچیدگی NP-Complete ها دشوارترین مسائل کلاس NP هستند و جزء مسائلی می‌باشند که احتمال حضورشان در کلاس P خیلی کم است. علت این امر این می‌باشد که اگر یک رامحل پیدا شود که بتواند یک مسئله NP-Complete را حل کند، می‌توان از آن الگوریتم برای حل کردن سریع همه مسائل NP-Complete استفاده کرد. به خاطر این مسئله و نیز بخاطر اینکه تحقیقات زیادی برای پیدا کردن الگوریتم کارآمدی برای حل کردن اینگونه مسائل با شکست مواجه شده‌اند، وقتی که مسئله‌ای به عنوان NP-Complete معرفی شد، معمولاً اینطور قلمداد می‌شود که این مسئله در زمان Polynomial قابل حل شدن نمی‌باشد، یا به بیانی دیگر هیچ الگوریتمی وجود ندارد که این مسئله را در زمان Polynomial حل نماید. کلاس متشکل از مسائل NP-Complete با نام NP-C نیز خوانده می‌شود.

(۲۵) الگوریتم رام نشدنی کدام است؟

۱. الگوریتم هایی با مرتبه زمانی $2n.n$ و $3n$ را مسائل رام نشدنی می نامند.

۲. مسائلی که نوشتن یک الگوریتم کارآمد برای آنها غیرممکن است مسائل رام نشدنی میگویند.

۳. الگوریتم هایی که مرتبه زمانی آنها چند جمله ای باشد را مسائل رام نشدنی می نامند.

۴. الگوریتم هایی که مرتبه زمانی آنها $\log n.n$ باشد را مسائل رام نشدنی میگویند.

پاسخ) گزینه ب

مسائلی که نتوان برای آنها الگوریتمی با مرتبه زمانی چند جمله ای پیدا کرد ، مسائل رام نشدنی نامیده می شود
الگوریتمهایی با مرتبه زمانی $2n, 3n, n!$ یا هر الگوریتمی که مرتبه زمانی آن غیر چند جمله ای باشد (نمایی) را مسائل رام نشدنی نامند

سوالات تشریحی

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

فاکتوریل یک عدد صحیح n ، به صورت بازگشتی زیر قابل تعریف است:

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

حال تابع بازگشتی زیر را برای محاسبه فاکتوریل یک عدد به صورت بازگشتی ارائه میدهیم:

```
Int    fact ( int n )
{
    If ( n== 0)
        Return(1);
    Else
        Return( n * fact (n-1));
```

}

به ازای $n=3$ تابع دو بار فراخوانی میشود. بنابراین مقادیر فراخوانی شده اول در پشته سیستم ذخیره میشود و عمل فراخوانی دوباره ادامه می یابد تا اینکه $n=0$ شود

در اینصورت برای محاسبه عملیات لازم در توابع فراخوانی شده ، مقدار یک بازگشت داده میشود
به ازای هر مرحله بازگشت یک عمل حذف از بالای پشته انجام میگردد و عین حال عملیات لازم برای بازگشت بعدی انجام میگردد. تا زمانی که پشته خالی نشده باشد عمل بازگشت ادامه می یابد.

$$T(n) = T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + 1 \quad (۳)$$

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

حدس میزنیم که $T(n) \in O(\log n)$ باشد بنابراین باید عدد ثابت مثبت مانند C وجود داشته باشد بطوریکه:

$$T(n) \leq C \log n$$

عبارت بالا را با استقرا ثابت میکنیم:

پایه استقرا

$$n = 2 \Rightarrow T(2) \leq C \log 2 = C$$

$$\Rightarrow C \geq T(2) > 0$$

فرض استقرا: برای $k < n$ فرض میکنیم:

$$T(k) \leq C \log_2^k$$

حکم استقرا: باید ثابت کنیم که (به ازای هر n):

$$T(n) \leq C \log_2^n$$

$$\Rightarrow T(n) \leq C \log_2^{\left\lfloor \frac{n}{2} \right\rfloor} + 1$$

$$\leq C \log n - C + 1$$

$$\leq C \log n$$

که در آن باید $C \geq 1$ باشد. در اینصورت $T(n) \in O(\log n)$.

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

روال انتخاب حریصانه (Selection): در این گام یک عنصر برای اضافه شدن به مجموعه جواب انتخاب می شود. معیار یا روال انتخاب عنصر برای اضافه شدن، ارزش آن عنصر است. بسته به نوع مسئله هر عنصر ارزشی دارد که با ارزشترین آنها انتخاب می شود.

امکان سنجی و افزودن (Feasible): پس از انتخاب یک عنصر به صورت حریصانه، باید بررسی شود که آیا امکان اضافه کردن آن به مجموعه جواب های قبلی وجود دارد یا نه. گاهی اضافه شدن عنصر یکی از شرایط اولیه مسئله را نقض می کند که باید به آن توجه نمود. اگر اضافه کردن این عنصر هیچ شرطی را نقض نکند، عنصر اضافه خواهد

شد؛ وگرنه کنار گذاشته شده و بر اساس گام اول عنصر دیگری برای اضافه شدن انتخاب می‌شود. اگر گزینه دیگری برای انتخاب وجود نداشته باشد، اجرای الگوریتم به اتمام می‌رسد.

بررسی اتمام الگوریتم (Solution): در هر مرحله پس از اتمام گام ۲ و اضافه شدن یک عنصر جدید به مجموعه جواب، باید بررسی کنیم که آیا به یک جواب مطلوب رسیده‌ایم یا خیر؟ اگر نرسیده باشیم به گام اول رفته و چرخه را در مراحل بعدی ادامه می‌دهیم.

نیمه دوم ۹۳-۹۴

(۲) در الگوریتم زیر در صورتی که $m=n$ 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;

$$1. \quad o\left(\frac{m+1}{2}\right) \quad 2. \quad o(n^2) \quad 3. \quad o\left(\frac{m(m+1)}{2}\right) \quad 4. \quad o(n^3)$$

پاسخ) گزینه د

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

$$1. \quad 25 \quad 2. \quad 108 \quad 3. \quad 2592 \quad 4. \quad 3456$$

پاسخ) گزینه ب

(۶) مرتبه زمانی رابطه بازگشتی مقابل برابر است با :

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

$$1. \quad o(n^2) \quad 2. \quad o(n^{\log n}) \quad 3. \quad o(\log n) \quad 4. \quad o(n)$$

پاسخ) گزینه الف

(۸) چند مورد از عبارات زیر صحیح می‌باشد؟

-الگوی جستجو برای روش عقبگرد به صورت جستجو در پهنای می‌باشد.

-در روش انشعاب و تحدید روش جستجوی درخت به ترتیب عمق می‌باشد.

-در هر دو روش بازگشت به عقب و انشعاب و تحدید شاخه‌هایی از درخت هرس می‌شود.

$$1. \quad 3 \quad 2. \quad 2 \quad 3. \quad 1 \quad 4. \quad 0$$

پاسخ) گزینه ج

- الگوی جستجو برای روش عقبگرد به صورت جستجو در عمق می‌باشد.

- در روش انشعاب و تحدید یکی از روشهای جستجوی درخت ، جستجو به ترتیب پهنا (سطحی) می باشد.

۱۰) کدامیک از مرتبه زمانی های زیر جزو مسائل رام نشدنی نمیباشد؟

۱. 2^n ۲. 3^n ۳. n^4 ۴. $n!$

پاسخ) گزینه ج

مسائل رام نشدنی مسائلی اند که نتوان برای آنها الگوریتمی با مرتبه زمانی چند جمله ای پیدا کرد و الگوریتمهایی با مرتبه زمانی $2n, 3n, n!$ یا هر الگوریتمی که مرتبه زمانی آن غیر چند جمله ای باشد (نمایی).

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) گزینه ب

* حداکثر تعداد گره ها در سطح i ام یک درخت دودویی 2^{i-1} است *

حداکثر تعداد گره ها در یک درخت دودویی به عمق k ، $2^k - 1$ 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) گزینه الف

هافمن ناتوان در پیدا کردن کارآمدترین روش کد دودویی، تصمیم گرفت خود روشی را ابداع کند. روش هافمن، استفاده از درخت دودویی مرتب شده با استفاده از تکرار بود. که بجای ساخته شدن از بالا به پایین؛ از پایین به بالا ساخته می شد. انواع مختلفی از کدگذاری هافمن وجود دارد

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) گزینه ب

الگوریتم پریم برای ساخت درخت پوشای کمینه از یک گراف به کار می رود . درخت پوشای کمینه ، زیر گرافی از یک گراف موزون است که شامل تمام رئوس گراف و برخی یال های آن می باشد به طوری که مجموع اوزان آن درخت نسبت به سایر درخت های ممکن حداقل شود . (می دانیم که معادل عدد کاتالان می توان با N راس درخت ایجاد کرد .)

الگوریتم پریم در هر مرحله : یالی با کمترین وزن را انتخاب می کند به طوری که:

۱. کم ترین وزن را میان سایر یال ها داشته باشد .
۲. قبلا انتخاب نشده باشد
۳. با یال های انتخابی قبلی تشکیل دور ندهد (زیرا در این صورت درخت نیست .)

این الگوریتم در هر مرحله یک درخت ایجاد می کند

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

۱. روش تقسیم و حل ۲. حریصانه ۳. برنامه نویسی پویا ۴. عقبگرد
- پاسخ) گزینه ب

در روش حریصانه در هر لحظه، با توجه به عناصر داده ای مفروض، عنصری را که دارای ویژگی بهترین یا بهینه است (مانند کوتاه ترین مسیر، بالاترین ارزش، کمترین سرمایه گذاری، بیشترین سود و ...) انتخاب می کنند بدون این که انتخاب های قبلی ما بعدی را در نظر بگیرد ولی انتخاب های بهینه محلی همواره منجر به راه حل بهینه سراسری نمی شود. این روش انتخاب، منجر به ارائه يك الگوریتم ساده و کارآمد می شود. تعیین درخت های پوشالی مینیمم با استفاده از الگوریتم های پرایم، کروسکال محاسبه کوتاه ترین مسیر يك منبع با کاربرد الگوریتم دایجسترا ، مسأله زمان بندی مانند بهینه سازی زمان انتظار و سرویس به کاربران برای دسترسی به دیسک گردان ها در يك شبکه رایانه ای، تعیین حداکثر بهره برای مشتریان در يك زمان معین و مسأله کوله پشتی (کسری، صفرو يك) با استفاده از روش حریصانه قابل اجرا هستند.

۲۰) بکارگیری روش تقسیم و حل برای کداميك از مسئله هاي زیر مناسب نمیباشد؟

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

پاسخ) گزینه الف

زمانی که مسأله را به چند زیرمسأله تقسیم می کنیم، اگر تقسیم طوری باشد که هر زیرمسأله خودش نزدیک به n ورودی داشته باشد، الگوریتم کارا نخواهد بود. نمونه چنین مسائلی محاسبه بازگشتی جمله n ام دنباله فیبوناتچی است.

۲۲) بدترین حالت الگوریتم Quick sort Quick sort چه زمانی رخ میدهد ؟

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) گزینه ج

این الگوریتم زمانی در بدترین حالت رخ میدهد که تمام pivot هایی که انتخاب میشوند در هر مرحله بزرگترین یا کوچکترین عنصر باشند (یکی از جاهایی که رخ میدهد زمانی هست که آرایه کاملاً مرتب باشد) برای بهتر انتخاب کردن pivot و جلوگیری از بدترین حالت از روش های آماری استفاده می کنند برای مثال pivot روش هر مرحله بصورت میانه ۳ تا عنصر از این آرایه در نظر می گیرند.

۲۴) زمان یک جستجوی موفق در بدترین حالت در الگوریتم جستجوی دودویی برابر است با:

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

پاسخ) گزینه الف

عملیات اصلی بر روی یک درخت جستجوی دودویی به زمانی متناسب با ارتفاع درخت احتیاج دارد. برای یک درخت دودویی کامل با n گره چنین عملیاتی در بدترین حالت در زمان $O(\log n)$ اجرا میشود. بنابراین اگر درخت یک زنجیر خطی با n گره باشد همین عملیات در زمان بدترین حالت $O(n)$ اجرا میشود. امید ریاضی ارتفاع یک درخت جستجوی دودویی که به تصادف ساخته شده است برابر با $O(\log n)$ است. بنابراین عملیات اصلی مجموعه پویا بر روی چنین درختی در حالت میانگین به زمان $O(\log n)$ احتیاج دارد.

سوالات تشریحی

۲) فرض کنید لیستی حاوی عناصر زیر باشد:

17,20,10,25,11,8,18

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

__ عنصر محور(اولین عنصر): ۱۷

8-10-11 ۱۷ 18-20-25

عناصر کوچکتر از عنصر محور سمت چپ و عناصر بزرگتر از عنصر محور سمت راست قرار گرفته است و به صورت غیر نزولی مرتب شده

مرحله اول:

تابع تقسیم بندی با مقدار زیر فراخوانی میشود:

Partition(0,7,0)

خروجی تابع partition:

8-10-11 ۱۷ 20-25-18

و مقدار pivotpoint برابر با ۳ که به عنوان خروجی ارسال میشود . در این مرحله تابع QuickSort به صورت زیر فراخوانی میشود:

QuickSort(0,2)

مرحله دوم:

تابع تقسیم بندی با مقادیر زیر فراخوانی میشود:

Partition(0,7,0)

خروجی تابع partition:

8-10-11

در اینجا ۸ عنصر محور میباشد

و مقدار pivotpoint برابر صفر بوده و به عنوان خروجی ارسال میشود.

در این مرحله تابع QuickSort به صورت زیر فراخوانی میشود

QuickSort(0,-1)

مرحله سوم:

تابع QuickSort با مقادیر زیر فراخوانی میشود:

QQuickSort(1,2)

در این صورت تابع تقسیم بندی با مقادیر زیر فراخوانی میشود:

Partition(1,2,0)

خروجی تابع partition:

۱۰ ۱۱

عنصر محور: ۱۰

و مقدار pivotpoint برابر ۱ بوده و به عنوان خروجی ارسال میگردد. در این مرحله تابع QuickSort با مقادیر زیر فراخوانی میشود

QuickSort(1,0)

مرحله چهارم:

ابع QuickSort با مقادیر زیر فراخوانی میشود:

QuickSort(2,2)

مرحله پنجم:

تابع QuickSort با مقادیر زیر فراخوانی میشود:

(مقادیر لیست از اول تا عنصر محور تا حالا مرتب شده اند)

QuickSort(4,7)

در اینصورت تابع تقسیم‌بندی با مقادیر زیر فراخوانی خواهد شد:

Partition(4,7,0)

خروجی تابع Partition:

18-20-25

عنصر محور: ۲۰

همچنین pivotpoint را برابر ۵ قرار داده و به تابع QuickSort با مقادیر زیر فراخوانی میشود:

QuickSort(4,4)

مرحله ششم:

تابع QuickSort با مقادیر زیر فراخوانی خواهد شد:

Partition(6,7,0)

خروجی Partition:

۲۳ . ۲۵

و مقدار pivotpoint برابر ۷ بوده و به عنوان خروجی ارسال میگردد.

بنابراین کل خروجی بعد از انجام عملیات ، به صورت زیر خواهد بود:

8-10-11-17-18-20-25

و این خروجی نهایی تابع QuickSort با مقادیر ۰ و ۷ و لیست S با مقادیر بیان شده، خواهد بود که یک لیست مرتب غیرنزولی میباشد.

(۴) الگوریتم فلویید را نوشته و پیچیدگی زمانی این الگوریتم را به دست آورید.

$$D[v_2][\emptyset] = 5$$

$$D[v_3][\emptyset] = 6$$

$$D[v_4][\emptyset] = 8$$

$$D[v_2][\{v_3\}] = 15$$

$$D[v_2][\{v_4\}] = 18$$

$$D[v_3][\{v_2\}] = 18$$

$$D[v_3][\{v_4\}] = 20$$

$$D[v_4][\{v_2\}] = 13$$

$$D[v_4][\{v_3\}] = 15$$

$$D[v_2][\{v_3, v_4\}] = 25$$

$$D[v_3][\{v_2, v_4\}] = 25$$

$$D[v_4][\{v_3, v_2\}] = 23$$

$$D[v_1][\{v_3, v_4, v_2\}] = 35$$