

گراف

سوالات سری اول:

۱- گزینه ۳:

این تابع زمانی مربوط به الگوریتم مرتب سازی ادغامی است که در این روش برای مرتب سازی n عنصری آنها را به دو زیر آرایه ی $\frac{n}{۲}$ عنصری تقسیم می کنیم و هر دو زیر آرایه را مرتب می کنیم و در نهایت این دو زیر آرایه با یکدیگر merge می شوند.

۳- گزینه ۱:

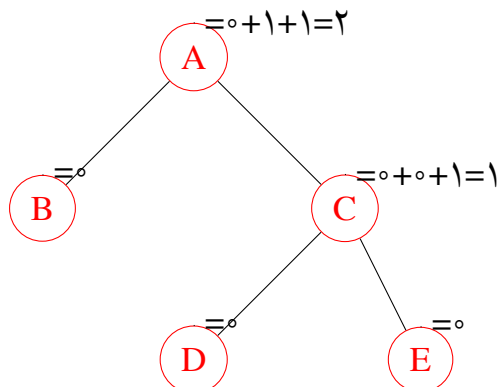
این تابع بصورت بازگشتی، به تعداد b بار، a را در خودش ضرب می کند که معادل است با a^b .

۵- گزینه ی ۲:

$$\begin{aligned} F(۳, ۶) &= F(۲, ۶) + F(۲, ۵) \\ &= F(۱, ۶) + F(۱, ۵) + F(۱, ۵) + F(۱, ۴) \\ &= ۱ + ۱ + ۱ + ۱ \\ &= ۴ \end{aligned}$$

۷- گزینه ۱:

برای گره های E و D مقدار این تابع برابر با صفر است و برای گره های C و A مقدار این تابع برابر است با مجموع مقدار زیرشاخه های چپ و راست به اضافه ی ۱.



۹- گزینه ی ۱:

$$T(n) = ۳T\left(\frac{n}{۴}\right) + \underbrace{n}_{f(n)} \xrightarrow{\text{قضیه اصلی (master)}} n^{\log_4 3} = n^{\log_4 3}$$

از آنجایی که $f(n) = n > n^{\log_2 3}$ نتیجه می گیریم که مرتبه ی زمانی این تابع برابر با $f(n)$ یا همان n است.

۱۱- گزینه ۳:

در روش جستجوی دودویی برای این لیست، ابتدا عنصر وسط این لیست یعنی عدد ۲۳ را با ۱۸ مقایسه می کنیم و چون ۱۸ از کوچکتر است به نیمه ی سمت چپ این لیست می رویم. حال عنصر وسط نیمه ی سمت چپ برابر با عدد ۱۸ است که همان عنصری است که دنبال آن بودیم، یعنی با دو مقایسه به عنصر مورد نظر رسیدیم.

۱۳- گزینه ی ۲:

پیچیدگی زمانی الگوریتم مرتب سازی سریع در بدترین حالت برابر با $\Theta(n^2)$ است که مربوط به زمانی است که لیست ورودی از قبل مرتب شده باشد.

۱۵- در روش کروسکال برای این گراف، ابتدا یال ۱ و سپس یال ۱۲ و سپس یال ۲۳ که به ترتیب دارای کمترین وزن هستند انتخاب می شوند. لذا گزینه ی ۴ صحیح است.

۱۷- گزینه ی ۳:

ابتدا ماتریس های C و B که به ترتیب دارای 3×3 ستون و 3×3 سطر هستند در یکدیگر ضرب می شوند که حاصل این ضرب، یک ماتریس 12×2 است. حال این ماتریس در ماتریس D ضرب می شود که حاصل این ضرب، یک ماتریس 8×2 است و در نهایت این ماتریس در ماتریس A ضرب می شود تا کمترین تعداد عمل ضرب انجام شود.

۱۹- گزینه ۳:

این گزینه اشتباه است زیرا درخت تصمیم در تکنیک عقبگرد نیز کاربرد دارد.

۲۱- گزینه ی ۳:

روش انتخاب و تحدید، شکل بهبودیافته ای از روش تقسیم و حل نمی باشد لذا عبارت ”ب” اشتباه می باشد و عبارت های ”الف” و ”ج” در مورد روش انتخاب و تحدید صحیح می باشند لذا گزینه ی ۳ صحیح است.

۲۳- گزینه ی ۲:

در این گراف فاصله ی بین گره های ۲ و ۱ برابر با ۹ است و همچنین فاصله ی بین گره های ۲ و ۵ برابر با بی نهایت است زیرا هیچ یال مستقیمی بین این گره وجود ندارد. و این فاصله ها فقط در ماتریس گزینه ی ۲ به درستی نوشته شده اند.

۲۵- گزینه ۳:

برای مسئله رنگ آمیزی گراف، هیچ الگوریتمی با مرتبه ی زمانی چندجمله ای وجود ندارد و جزو مسائل Np-Complete می باشد. برای سایر گزینه ها، الگوریتم هایی با مرتبه ی زمانی چندجمله ای وجود دارد.

۱- از روش حل رابطه ی بازگشتی همگن استفاده می کنیم:

$$T(n) = \begin{cases} 3T(n-1) + 4T(n-2) & \text{if } n \geq 2, \\ t(0) = 0, t(1) = 1. \end{cases}$$

$$T(n) = 3T(n-1) + 4T(n-2) \Rightarrow T(n) - 3T(n-1) - 4T(n-2) = 0 \xRightarrow{\text{مقدار مشخصه}} r^2 - 3r - 4 = 0$$

$$\Rightarrow (r-4)(r+1) = 0 \Rightarrow \begin{cases} r_1 = 4, \\ r_2 = -1. \end{cases} \Rightarrow T(n) = C_1 4^n + C_2 (-1)^n$$

$$T(0) = 0 \rightarrow C_1 4^0 + C_2 (-1)^0 = 0 \rightarrow C_1 + C_2 = 0$$

$$T(1) = 1 \rightarrow 4C_1 - C_2 = 1 \rightarrow 4C_1 - C_2 = 0$$

$$\Rightarrow \begin{cases} C_1 + C_2 = 0, \\ 4C_1 - C_2 = 1. \end{cases}$$

$$\Rightarrow 4C_1 = 1 \rightarrow \boxed{C_1 = \frac{1}{4}}, \boxed{C_2 = -\frac{1}{4}}$$

$$\rightarrow \boxed{T(n) = \frac{1}{4} 4^n - \frac{1}{4} (-1)^n}$$

مرتبه اجرایی این رابطه ”نمایی” است.

۳- با استفاده از روش حریصانه، کاری را که دارای بیشترین بهره است را انتخاب می کنیم و انجام آن کار را تا آخرین لحظه ی ممکن به تعویق می اندازیم یعنی در این سوال، ابتدا کار شماره ۱ که دارای بهره ۶۰ است را برای انجام انتخاب می کنیم و انجام آن کار را به لحظه ی ۳ تخصیص می دهیم. حال کار بعدی برای انجام را انتخاب می کنیم که در این مرحله، کار شماره ۲ برای لحظه ی ۱ انتخاب می شود و در نهایت کار شماره ۴ برای لحظه ی ۲ انتخاب می شود:

مهلت	۱	۲	۳
شماره کار	۲	۴	۱
بهره	۵۰	۲۰	۶۰

$$\text{سود نهایی} = ۱۳۰ = ۵۰ + ۲۰ + ۶۰$$

۵- برای حل این مسئله با روش پویا از تعاریف زیر استفاده می کنیم

مجموعه همه ی رئوس گراف V

زیرمجموعه ای از رئوس گراف A

طول کوتاه ترین مسیر از V_i به V_1 که از هر راس زیرمجموعه A دقیقاً یکبار می گذرد $D[V_i][A]$

در این سوال اگر $A = \{V_3\}$ باشد داریم:

$$D[V_2][A] = \text{len}[V_2, V_3, V_1] = \infty$$

ماتریس همجواری این گراف بصورت مقابل است:

$$\begin{array}{c} V_1 \quad V_2 \quad V_3 \quad V_4 \\ \begin{array}{c} V_1 \\ V_2 \\ V_3 \\ V_4 \end{array} \begin{bmatrix} 0 & 2 & 9 & 6 \\ 1 & 0 & 6 & 4 \\ \infty & 7 & 0 & 8 \\ \infty & 3 & \infty & 0 \end{bmatrix} \end{array}$$

و با استفاده از روش برنامه نویسی پویا، قدر بهینه ای که برای این گراف بدست می آید برابر است با:

$$[V_1, V_3, V_4, V_2, V_1] = 21$$

سوالات سری دوم:

۲- گزینه ۴:

در این الگوریتم حلقه ی اول به اندازه ی n بار اجرا می شود ولی تعداد اجرای حلقه دوم وابسته به مجموع عناصر آرایه ی A در اندیس i است، لذا مرتبه ی اجرائی این الگوریتم برابر با $O(n+m)$ است.

۴- گزینه ۳:

از آنجایی که حاصل $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)}$ برابر با یک عدد ثابت شده است می توان نتیجه گرفت که :

$$(f(n) \in \Theta(g(n)))$$

۶- گزینه ی ۳:

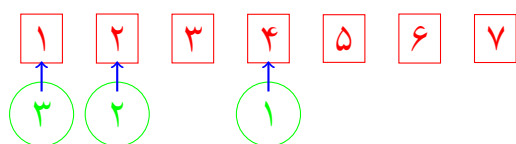
این رابطه ی بازگشتی را به ازای $f(a, 9)$ حل می کنیم:

$$\begin{aligned} f(a, 9) &= a[9] + f(a, 7) \\ &= 10 + 8 + f(a, 5) \\ &= 10 + 8 + 6 + f(a, 3) \\ &= 10 + 8 + 6 + 4 + f(a, 1) \\ &= 10 + 8 + 6 + 4 + 2 + f(a, -1) \\ &= 10 + 8 + 6 + 4 + 2 + 1 \\ &= 31 \end{aligned}$$

۸- گزینه ۴:

$$\begin{aligned}
 f(n) &= f(n-2) * f(n-2) = (f(n-2))^2 \rightarrow 2^{\frac{n}{2}} = 2 \\
 &= f(n-4) * f(n-4) * f(n-4) * f(n-4) = (f(n-4))^4 \rightarrow 2^{\frac{n}{4}} = 4 \\
 &= f(n-6) * f(n-6) * f(n-6) * f(n-6) * f(n-6) * f(n-6) \\
 &= (f(n-6))^6 \rightarrow 2^{\frac{n}{6}} = 8 \\
 \Rightarrow f(n) &\in o(2^{\frac{n}{6}})
 \end{aligned}$$

۱۰- گزینه ۲:



در یک آرایه ی ۷ عنصری مطابق آرایه ی مقابل ابتدا عنصر مورد نظر با عنصر میانی آرایه مقایسه می شود و اگر کوچکتر از عنصر میانی بود به سمت چپ آرایه می رویم و با عنصر میانی زیرآرایه ی چپ مقایسه می کنیم و اگر برابر نبوده باز به زیرآرایه ی سمت چپ می رویم که در مجموع برابر با ۳ مقایسه می شود. برای حالتی که عنصر موردنظر از عنصر میانی بزرگتر باشد به زیرآرایه سمت راست می رویم که باز هم نیاز به ۳ مقایسه است و در حالت متوسط به ۳ مقایسه در جستجوی ناموفق نیاز داریم.

۱۳- گزینه ۳:

تعداد عمل ضرب در روش پویا برابر با n^3 می باشد که این مقدار برای ضرب ماتریس های 8×8 برابر با $8^3 = 512$ عمل ضرب می باشد. در ضرب ماتریس ها به روش استراسن در مرحله ی اول نیاز به 343 ضرب و در مرحله نهایی نیاز به 49 ضرب است که در مجموع برابر با 392 ضرب می باشد.

۱۴- گزینه ۳:

با استفاده از روش هار حریصانه اگر بصورت جدول زیر کارها را انجام بدهیم، سود بدست آمده برابر با ۱۲۱ می شود که این میزان سود برابر با بیشترین سود ممکن برای این مسئله است:

شماره کار	۱	۷	۳	۱۰	۶	۲	۱۱
بازه زمانی	۱	۲	۳	۴	۵	۶	۷
سود	۱۹	۲۷	۲۵	۲۳	۱۶	۷	۴

۱۶- گزینه ۱:

این الگوریتم بیانگر رابطه ی محاسبه ی ضریب دوجمله ای است که به صورت زیر بیان می شود:

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$$

که این رابطه نیاز به $\binom{n}{k} - 1$ عمل جمع دارد.

۱۸- گزینه ۳:

حداکثر تعداد بازی زمانی اتفاق می افتد که تیم های A و B به ترتیب برنده شوند. به عنوان مثال در بازی اول، تیم A برنده شود و در بازی دوم تیم B برنده شود و در بازی سوم تیم A و در بازی چهارم

تیم B برنده شوند و این روال ادامه پیدا کند. در این صورت حداکثر بازی مورد نیاز برابر با $2n - 1$ می باشد.

۲۰- گزینه ۲:

می خواهیم تعداد درخت های جستجوی دودویی متفاوت با n کلید متمایز را بدست آوریم. فرض کنید که گره های هر کدام از این درخت ها را به روش میان ترتیب از ۱ تا n شماره گذاری می کنیم. با این شماره گذاری فرض کنید که شماره $n \geq i \geq 1$ ریشه درخت باشد. در آن صورت زیر درخت راست آن حتما باید $n - i$ گره داشته باشد.

حال اگر $T(n)$ تعداد درخت دودویی با n گره باشد، تعداد زیردرخت های چپ و راست ریشه به ترتیب برابر با $T(i - 1)$ و $T(n - i)$ خواهند بود. در آن صورت اگر گره i ریشه باشد، حاصلضرب $T(i)T(n - i)$ برابر تعداد کل درخت هاست و با در نظر گرفتن همه ی حالات، رابطه ی بازگشتی زیر بدست می آید که همان عدد کاتالان است:

$$\frac{1}{n+1} \binom{2n}{n}$$

۲۲- گزینه ۳:

روش عقبگرد به طور عمده در مسائل تصمیم گیری استفاده می شود، لذا گزینه ی ۳ صحیح است.

۲۴- گزینه ۲:

تعداد کل گره های درخت ??? حالت برای الگوریتم یافتن مدارهای همیلتونی با استفاده از تکنیک عقبگرد، با استفاده از رابطه ی $1 + (n - 1) + (n - 1)^2 + \dots + (n - 1)^{n-1}$ بدست می آید که این عبارت معادل است با:

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

سوالات تشریحی سری دوم:

از الگوریتم پریم برای حل این مسئله استفاده می کنیم: ابتدا از گره ۱ شروع می کنیم و از بین گره های ۲ و ۳ و ۴ که همگی دارای فاصله ی ۵ تا گره ۱ هستند، یکی را بصورت دلخواه انتخاب می کنیم. در اینجا ما گره شماره ۲ را انتخاب می کنیم و طول مورد نیاز تا این مرحله برابر با ۵ است. حال گره ای که کمترین فاصله تا مجموعه گره های $\{1, 2\}$ دارد را انتخاب می کنیم که گره ی شماره ۳ است و این گره را به این مجموعه اضافه می کنیم: $\{1, 2, 3\}$. تا به اینجا طول نوار مسی مورد نیاز برابر با ۱۰ است. حال گره ای که کمترین فاصله تا مجموعه ی $\{1, 2, 3\}$ دارد را انتخاب می کنیم که گره ی شماره ۴ است که دارای فاصله ۵ از گره شماره ۱ است و این گره به این مجموعه اضافه می شود: $\{1, 2, 3, 4\}$ و طول مورد نیاز نیز در مجموع برابر با ۱۵ می شود. در نهایت گره شماره ۵ که دارای فاصله ی ۵ با گره شماره ۴ است به این مجموعه اضافه می شود: $\{1, 2, 3, 4, 5\}$ و طول نهایی نوار مسی برابر با $15 + 5 = 20$ می شود.

۴- در این روش ابتدا تمامی آیتم ها را بر اساس $\frac{p_i}{w_i}$ مرتب می کنیم:

i	p_i	w_i	$\frac{p_i}{w_i}$
۵	۲۰	۱۵	۱/۳۳
۲	۱۰	۸	۱/۲۵
۴	۱۵	۲۵	۰/۶
۱	۸	۱۶	۰/۵
۳	۵	۱۵	۰/۳۳

در ابتدا کوله پشتی خالی است و وزن و ارزش قطعات برابر با صفر است:

$$\begin{cases} w = 0, \\ p = 0. \end{cases}$$

حال مطابق درخت زیر، قطعه شماره ۵ را به کوله پشتی اضافه می کنیم

$$\begin{cases} w = 15, \\ p = 20. \end{cases}$$

در مرحله بعد گره شماره ۲ به این درخت اضافه می شود

$$\begin{cases} w = 15 + 8 = 23, \\ p = 20 + 10 = 30. \end{cases}$$

حال اگر آیتم های ۳ یا ۴ به درخت اضافه شوند، وزن مجموع، از وزن مجاز بیشتر می شوند، در نتیجه این گره ها انتخاب نمی شوند و به آیتم ۲ برمی گردیم و آیتم شماره ۱ انتخاب می شود

$$\begin{cases} w = 39, \\ p = 38. \end{cases}$$

حال اگر گره های ۳ یا ۴ را به این درخت اضافه کنیم، وزن مجموع از حد مجاز خارج می شود، در نتیجه عقبگرد می کنیم و به آیتم ۵ بر می گردیم. حال می توانیم گره ۴ را اضافه کنیم و ادامه بدهیم ولی همانطور که در درخت زیر مشاهده می کنیم، بیشترین سود بدست آمده برابر با ۳۸ و وزن قطعات برابر با ۳۹ می شود:

