

عنوان درس: طراحی الگوریتم	دانشگاه: پیام نور تهران شمال
استاد: آقای علی رضوی	پاسخ سوالات فرد نیم سال اول ۹۴-۱۳۹۳
نام دانشجو: عباس بهروز وزیری	شماره دانشجویی: ۹۶۳۸۳۵۲۷۲

پاسخ سوال ۱: گزینه (ب) درست است. زیرا: می دانیم که  $T_1(n) \in o(n^2)$  بنابراین  $C_1$  و  $n_1$  وجود دارد که برای:

$$\forall n \geq n_1 \quad T_1(n) \leq C_1(n^2)$$

و همچنین  $T_2(n) \in o(n^2)$  بنابراین  $C_2$  و  $n_2$  وجود دارد که برای:

$$\forall n \geq n_1 \quad T_2(n) \leq C_2(n \log n)$$

$$\implies T_1(n) + T_2(n) \leq C_1(n^2) + C_2(n \log n) \implies T_1(n) + T_2(n) \in O(n^2)$$

پاسخ سوال ۳: گزینه (د)،  $T(n) = T(n-1) + T(n-2) + 1$ ، درست است زیرا: در مرحله فراخوانی، مقدار متغیرها در پشته *push* می شوند. بنابراین برای  $n=3$ ، اول  $fact(3)$  فراخوانی می شود. بازای  $n=3$  تابع دوباره فراخوانی می شود بنابراین مقادیر اول در پشته سیستم ذخیره می شود و عمل فراخوانی دوباره ادامه می یابد تا اینکه  $n=1$  شود. در این صورت برای محاسبه عملیات لازم در توابع فراخوانی شده، مقدار یک بازگشت داده می شود. بازای هر مرحله بازگشت یک عمل حذف از بالای پشته انجام می گیرد و در عین حال عملیات لازم برای بازگشت بعدی انجام می گیرد تا زمانی که پشته به خانه ۱ برسد عمل بازگشت ادامه می یابد.

پاسخ سوال ۵: گزینه (ج) درست است زیرا:

```

if(tree == NULL)
    return 0;
else {
    inli = f(tree → left);
    inli = f(tree → right);  $\implies$  خروجی = ۴
    if(i > j)
        return ۱ + i;
    else
        return ۱ + j;
}

```

پاسخ سوال ۷: گزینه (ج) درست است زیرا:

(۱) ۱۱ ۱۲ ۱۸ ۲۰ ۲۱ ۲۳ ۲۷ ۴۰ ۷۵ ۸۰ ۸۵

عنصر میانی عدد

نصر میانی عدد ۲۳ هست که با ۴۰ برابر نیست چون  $۴۰ > ۲۳$  پس باید به زیر آرایه سمت

چپ چپ ۲۷ ۴۰ ۷۵ ۸۰ ۸۵ (۲)

(۳)  $۴۰ < ۷۵ \implies ۲۷ < ۴۰.۵$  (۴)

پاسخ سوال ۹: گزینه (د) درست است زیرا: در مرحله اول مرتب سازی سریع عناصر کوچکتر از

عنصر محور را در سمت چپ عناصر بزرگتر را در سمت راست لیست قرار می دهیم:

۳	۸	۷	۶	۹	۱۵	۱۰
---	---	---	---	---	----	----

پاسخ سوال ۱۱: گزینه (ج) درست است زیرا: اگر  $n \leq ۲$  باشد،  $\max, \min$  را پیدا می کنیم و برای  $n > ۲$ ، مسئله را به دو زیرمسئله با طول های  $\frac{n}{۳}$  تقسیم می کنیم و آنها را حل می کنیم نتایج حاصل از حل زیرمسئله را برای حل مسئله اصلی باهم ترکیب می کنیم با  $۲ - \frac{۳n}{۳}$  مقایسه بین اعداد ذخیره شده در این خانه ها بدست خواهد آمد.

پاسخ سوال ۱۳: گزینه (ب) درست است زیرا: طبق الگوریتم، ابتدا رأس  $Y = \{v_0\}$  انتخاب می شود و  $F = \phi$  خواهد بود.

مرحله اول تمام رئوس مجاور  $v_0$  را پیدا می کنیم. بنابراین خواهیم داشت:

می‌شود و داریم:  $f = \{e_{01}\}$ . به‌وضوح رئوس  $v_1$  رأس انتهایی خواهد بود و یال  $e_{01}$  به  $f$  اضافه می‌کنیم.

مرحله دوم: حال رئوس مجاور به  $Y$  را انتخاب می‌کنیم، نزدیک‌ترین رأس که  $V_2$  بوده انتخاب می‌کنیم.

مرحله سوم: نزدیک‌ترین رأس به  $Y$  رأس  $V_5$  بوده.

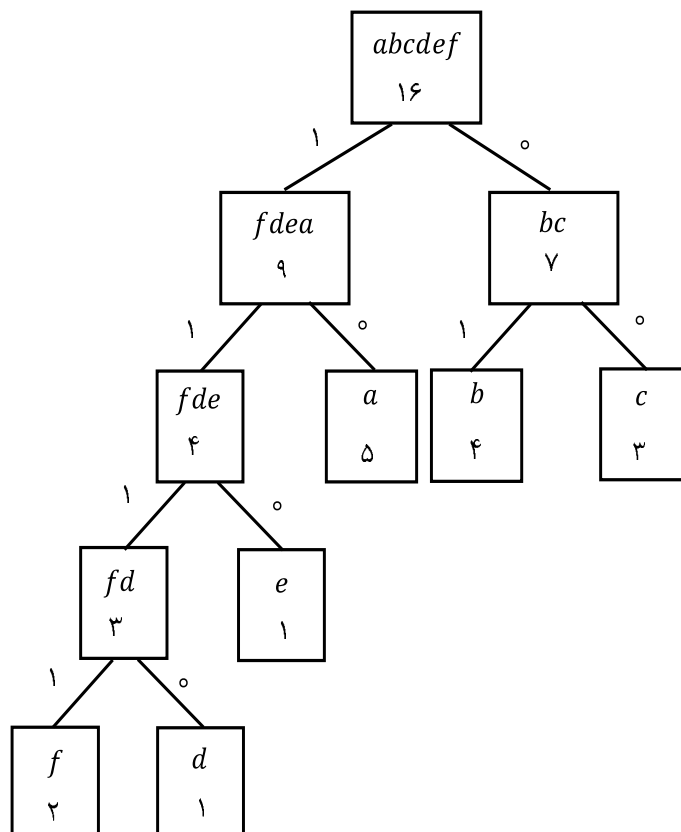
مرحله چهارم: نزدیک‌ترین رأس به  $Y$  رأس  $V_4$  بوده.

مرحله پنجم: نزدیک‌ترین رأس به  $Y$  رأس  $V_5$  بوده.

مرحله ششم: چون  $v = Y$  می‌باشد، بنابراین گراف حاصل به‌عنوان درخت پوشای مینیمم با مقدار هزینه‌ای می‌باشد.

پاسخ سوال ۱۵: گزینه (د) درست است زیرا:  $abcabbaccaabdfc$  (از هر صفریا یک در هر سطح رد شویم یک بیت می‌شود).

$$\text{بیت } 38 = (5 \times 2) + (4 \times 2) + (3 \times 2) + (1 \times 4) + (1 \times 3) + (2 \times 4)$$



پاسخ سوال ۱۷: گزینه (د) درست است زیرا:  $A_{۱۳ \times ۵}$ ,  $B_{۵ \times ۸۹}$ ,  $C_{۸۹ \times ۳}$ ,  $D_{۳ \times ۳۴}$

باید سعی کنیم اعداد بزرگی مانند ۸۹ و ۳۴ تنها یک بار استفاده شوند، چون ۳۴ از ابعاد وسطی است می توان با یک ضرب آنرا از بین برد.

$$M = A \times B \times C \times D \longrightarrow m = (A((BC)D))$$

در داخلی ترین ضرب  $B_{۵ \times ۸۹} * C_{۸۹ \times ۳}$  تعداد ضرب ها برابر است با  $۵ \times ۸۹ \times ۳$ . حالا ماتریس حاصل از ضرب  $BC$  یک ماتریس  $۵ \times ۳$  است و وقتی در  $D$  ضرب شود، تعداد ضرب های آن برابر است با  $۳۴ \times ۳ \times ۵$  ماتریس حاصل تا اینجا کار  $۳۴ \times ۵$  خواهد بود که برای ضرب  $A$  در آن به  $۳۴ \times ۵ \times ۱۳$  ضرب نیاز است، پس تعداد کل ضرب ها برابر است با: ۲۸۵۶

پاسخ سوال ۱۹: گزینه (الف) درست است زیرا: با  $n$  عنصر متمایز می توانیم  $\frac{1}{n!} \binom{2n}{n}$  درخت جست و جوی دو دویی بسازیم که همان عدد کاتالان است.

پاسخ سوال ۲۱: گزینه (ج) درست است زیرا: در حالت کلی دو وزیر  $a_{ij}$  و  $a_{kl}$  مورد حمله یکدیگر خواهند بود اگر یکی از سه حالت زیر رخ دهد:

$$(۱) \quad j = ۱ \text{ دو وزیر در یک ستون باشند.}$$

(۲)  $i - j = k - ۱$  یا  $i + j = k + ۱$  برای نوشتن برنامه  $n$  وزیر ابتدا تابع  $promising$  را طراحی می کنیم که وظیفه چک کردن سه شرط گفته شده را دارد. اگر هیچ کدام از سه شرط فوق برقرار نباشد خروجی الگوریتم یک وگرنه صفر خواهد بود.

پاسخ سوال ۲۳: گزینه (د) درست است زیرا: از بین مهره های موجود در شکل که توسعه نیافته اند گره ای را که دارای کمترین ارزش است ادامه می دهیم.

پاسخ سوال ۲۵: گزینه (الف) درست است زیرا: تعریف  $NP$  به درستی ذکر شده است، اما در گزینه های دیگر به درستی ذکر نشده است چرا که مسائلی که نوشتن یک الگوریتم کارآمد برای آنها غیر ممکن است، مسائل رام نشدنی می گویند. مسائلی که الگوریتم کارا برای آنها ابلاغ شده است ولی غیر ممکن بودن آنها نیز هنوز به اثبات نرسیده را مسائل  $NP$  کامل می گویند.

## تحلیل سوالات فرد نیم سال اول ۹۴-۹۳ (سوالات تشریحی)

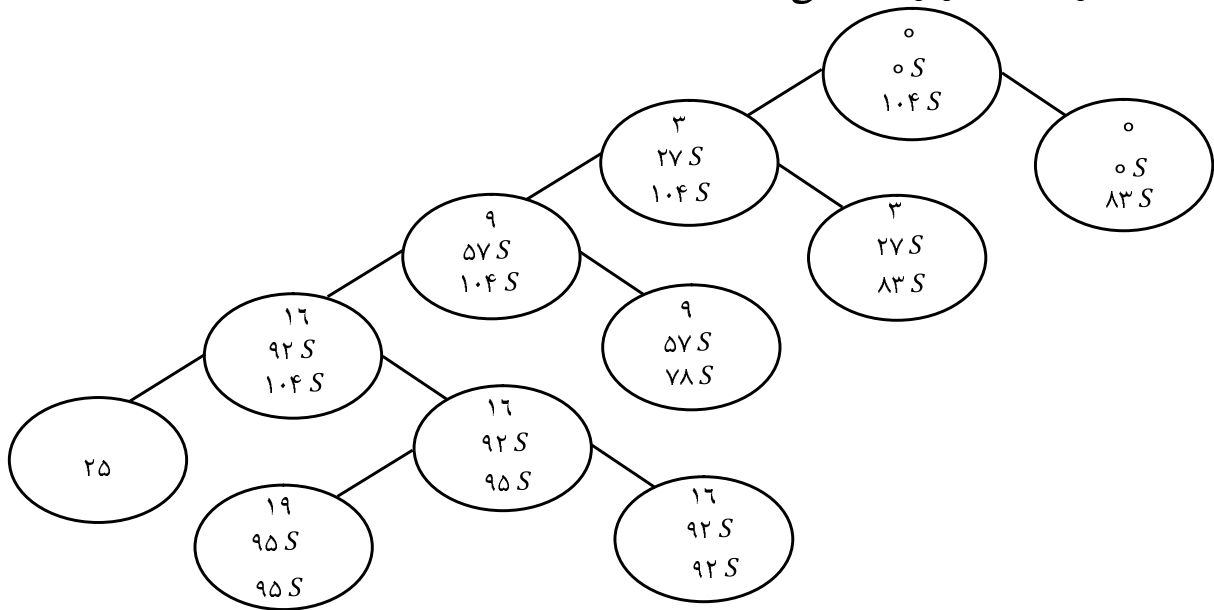
$$\{e_{12}, e_{21}, e_{41}, e_{16}, e_{35}\}$$

$$\begin{bmatrix} \circ & \circ/7 & 1/1 & 1/4 \\ & \circ & \circ/2 & \circ/4 \\ & & \circ & \circ/1 \\ & & & \circ \end{bmatrix}$$

.۱

.۳

۵. سود بیشینه برابر با ۹۵ می باشد.



## سوالات زوج

۲. (ج) فصل اول

ترتیب مرتبه زمانی از کوچک تر به بزرگ تر: ( $a$  عدد ثابت است).

$$o(1) < o(\sqrt{\log n}) < o(\sqrt{\log_2 n}) < o(n) < o(n \log n) < o(n^2) < o(n^3) < o(n^a)$$

$$< o\left(\left(\frac{3}{2}\right)^n\right) < o(n^2) < o(a^n) < o(n!) < o(n^n)$$

در چند جمله ای داده شده جمله  $5n \log n$  دارای بالاترین مرتبه زمانی است، پس پیچیدگی

زمانی برابر می شود با  $o(n \log n)$ .

۴. (ب) فصل اول

تعداد ضرب‌های حاصل ضرب دو ماتریس  $n \times n$  برابر می‌شود با  $n \times n \times n$ ، لذا مرتبه زمانی آن برابر با  $n^3$  می‌شود.

## ۶. (الف) فصل دوم

تابع سری فیبوناچی به صورت زیر است:

$$F(n) := \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ F(n-1) + F(n-2) & \text{if } n > 1 \end{cases}$$

$$fib(4) = fib(3) + fib(2)$$

$$fib(3) = fib(2) + fib(1)$$

$$fib(2) = fib(1) + fib(0)$$

$$fib(1) = 1$$

$$fib(0) = 0$$

$$fib(2) = fib(1) + fib(0) = 1 + 0 = 1$$

$$fib(3) = fib(2) + fib(1) = 1 + 1 = 2$$

$$fib(4) = fib(3) + fib(2) = 2 + 1 = 3$$

$$fib(4) + fib(3) = 3 + 2 = 5$$

## ۸. (ب) فصل سوم

تابع بازگشتی به صورت  $T(n)$  بیان می‌گردد. در گزینه‌های ۱، ۳ و ۴، پرانتز  $T$  براساس  $n$  بیان نشده است، لذا صحیح نیستند.

## ۱۰. (ج) فصل چهارم

	بهترین حالت	حالت میانگین	بدترین حالت
جستجوی موفق	$\Omega(1)$	$\theta(\log n)$	$o(\log n)$
جستجوی ناموفق	$\Omega(\log n)$	$\theta(\log n)$	$\theta(\log n)$

## ۱۲. (ب) فصل چهارم

در ماتریس استراسن:

$$\left\{ \begin{array}{l} P = (A_{11} + A_{22})(B_{11} + B_{22}) \\ Q = (A_{21} + A_{22})B_{11} \\ R = A_{11}(B_{12} - B_{22}) \\ S = A_{22}(B_{21} - B_{11}) \\ T = (A_{11} + A_{12})B_{22} \\ U = (A_{21} - A_{11})(B_{11} + B_{12}) \\ V = (A_{12} - A_{22})(B_{21} + B_{22}) \end{array} \right.$$

۱۴. (ب) فصل چهارم

اعداد صحیح بزرگ نصف می‌شوند و اگر  $n$  رقم داشته باشند، از وسط به دو عدد تبدیل

می‌شوند، مثلاً  $U$  به  $x$  و  $y$  تبدیل می‌شود و  $m = \left\lfloor \frac{n}{2} \right\rfloor$ .

۱۶. (ب) فصل پنجم

پیچیدگی زمانی الگوریتم پریم  $O(n^2)$  می‌باشد.

۲۰. (د) فصل ششم

$$D_0 = \begin{bmatrix} 0 & 1 & \infty & 1 & 5 \\ 9 & 0 & 3 & 2 & \infty \\ \infty & \infty & 0 & 4 & \infty \\ \infty & \infty & 2 & 0 & 3 \\ 3 & \infty & \infty & \infty & 0 \end{bmatrix}$$

$D^0$  همان ماتریس مجاورت گراف می‌باشد.

برای  $D^1$  داریم: ابتدا سطر و ستون ۱ را مانند جدول قبل قرار می‌دهیم.

$$D^1(2, 2) = 0, \quad D^n(2, 2) = D^n(3, 3) = 0$$

$$D^1(2, 3) = \min(D^0(2, 1) + D^0(1, 3), D^0(2, 3)) = \min(9 + \infty, 3) = 3$$

منظور از  $D^1(2, 4)$  این است که کوتاهترین مسیر از گره ۲ به گره ۴ با گره واسطه ۱ کدام است.

$$D^1(2, 4) = \min(D^0(2, 1) + D^0(1, 4), D^0(2, 4)) = \min(9 + 1, 2) = 2$$

$$D_1 = \begin{bmatrix} 0 & 1 & \infty & 1 & 5 \\ 9 & 0 & 3 & 2 & 14 \\ \infty & \infty & 0 & 4 & \infty \\ \infty & \infty & 2 & 0 & 3 \\ 3 & \infty & \infty & \infty & 0 \end{bmatrix}$$

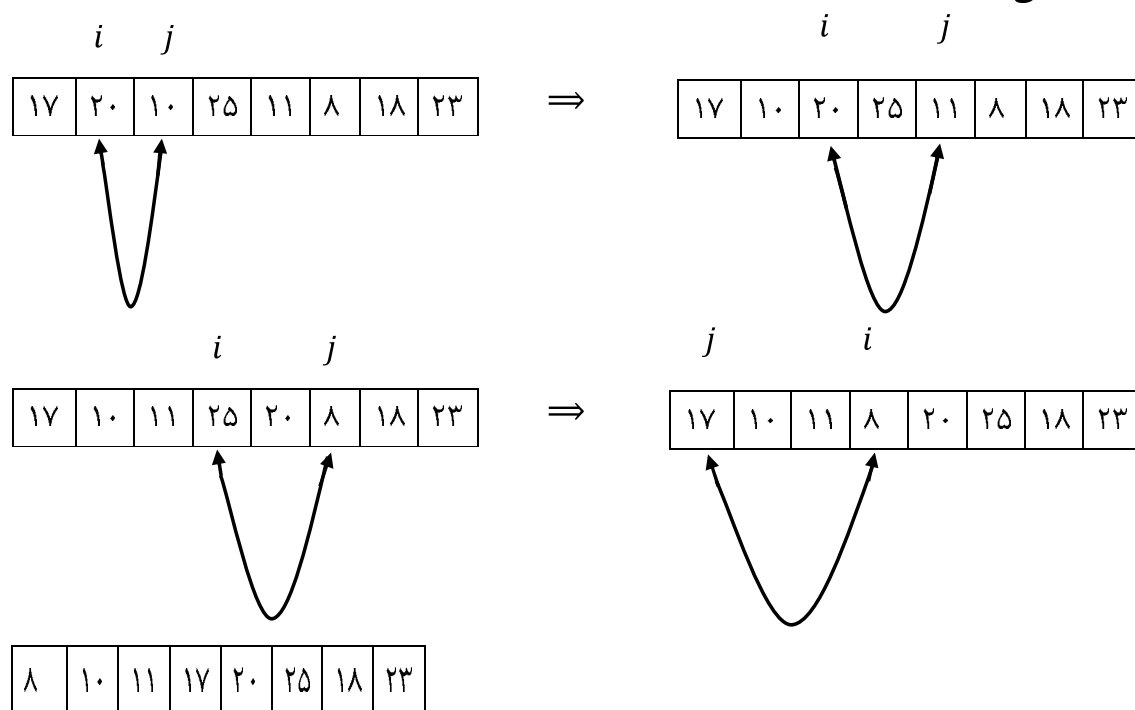
۲۲. (ج) فصل هفتم

در مسئله حاصل جمع زیرمجموعه‌ها می‌بایست تمام زیرمجموعه‌هایی از قطعات را بیابیم به‌طوری‌که مجموع اوزان آنها به اندازه وزن کوله یعنی ۲۱ باشد.

تشریحی :

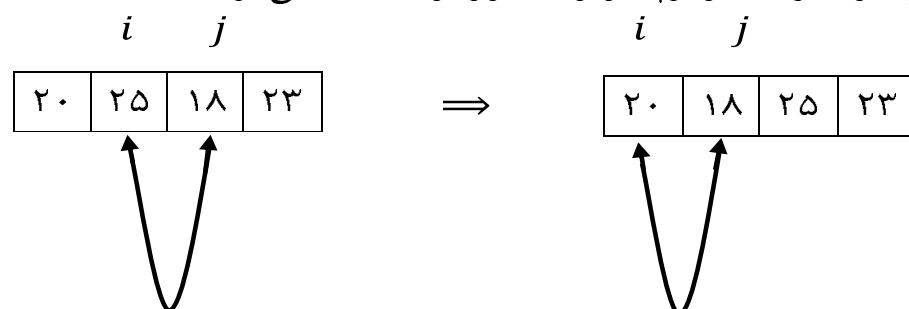
## ۲- فصل چهارم

ابتدا  $j$  در عناصر کوچکتر از ۱۷ متوقف شده و سپس  $i$  در عنصر بزرگتر از ۱۶ و پس از آن عناصر تعویض می‌شوند. اگر  $j$  کل آرایه را پیمایش کند و عنصر کوچکتر نباشد محل  $i$  با عنصر اول جابجا می‌شود.



در این حالت عنصر ۱۷ در محل خود در آرایه مرتب شده قرار می‌گیرد در مرحله بعد، همین اعمال برای آرایه سمت چپ و راست ۱۷ تکرار خواهد شد. برای آرایه سمت راست ۱۷:

$i$  در عنصر بزرگتر از ۲ و  $j$  در عناصر کوچکتر از ۲۰ قرار گرفته جابجا می‌شوند.





۱۸	۲۰	۲۵	۲۳
----	----	----	----

عدد ۲۰ در محل خود قرار می‌گیرد و این اعمال برای آرایه سمت راست و چپ آن تکرار خواهد شد.

۴- در الگوریتم پریم، در هر بار رأس جدیدی به مجموعه اضافه می‌گردد با این شرط که این رأس با وزن یا کمتری به مجموعه وصل شود. ابتدا  $V_0$  را در نظر می‌گیریم از مجموعه رئوس که به آن متصل است رأس  $V_1$  با کمترین وزن انتخاب می‌شود، سپس از مجموعه رئوس دیگری که به رئوس  $V_0$  و  $V_1$  متصلند  $V_2$  با کمترین وزن انتخاب می‌شود رأس سوم  $V_2$  خواهد بود و به همین ترتیب . . .

