عنوان درس: طراحی الگوریتم	دانشگاه: پیام نور تهران شمال
استاد: آقای علی رضوی	پاسخ سوالات نیم‌سال اول ۹۵–۱۳۹۴
نام دانشجو: مژگان متقی	شماره دانشجویی: ۹۷۰۱۵۳۴۰۳

پاسخ سوال ۱: **پیچیدگی قطعه مقابل کدام است؟**

$(1)\ For(i = 0;\ i < n;\ i++)$

$(2)\quad For(j = i;\ j < \min(i, k);\ j++)$

$(3)\qquad X++$

پاسخ صحیح گزینه سوم یعنی $o(n)$ می‌باشد چون با فرض $min(i, k) = m, m \in \mathbb{N}$ خواهیم داشت:

سطر	هزینه	تعداد اجزاء
۱	C_1	$n + 1$
۲	C_2	$(m - 1) + (m - 2) + ... + (m - n + 1)$
۳	C_3	$(m - 2) + (m - 3) + ... + (m - n)$

لذا:

$$T(n) = C_1 + C_1 n + C_2((m - 1) + (m - 2) + ... + (m - n + 1)) + C_3((m - 2) + (m - 3) + ... + (m - n))$$

$$T(n) \Longrightarrow C = \max(C_1, C_2, C_3) \Longrightarrow$$

$$\begin{aligned} T(n) &\le C(1 + n + (m - 1) + (m - 2) + ... + (m - n + 1) \\ &+(m - 2) + (m - 3) + ... + (m - n)) \\ &\le C(1 + n + 2m(n - 2)) = C((2m + 1)n - 4) \end{aligned}$$

با فرض $k = 2m + 1$ و m عددی کمتر از n و عددی مثبت خواهیم داشت:

$$T(n) \le C(kn - 4) \Longrightarrow T(n) \in o(n)$$

پاسخ سوال ۳: **کدام‌یک از موارد زیر در مورد پیچیدگی زمانی الگوریتم صحیح نیست؟**

رابطه غیر صحیح روابط ۱–۳–۴ می‌باشد چون داریم:

$$if\left(\lim\frac{T(n)}{g(n)}=+\infty\right)\Longrightarrow g(n)\in o(T(n))\ ,\ T(n)\notin\theta(g(n))$$

و مابقی روابط صحیح نمی‌باشند.

پاسخ سوال ۵: **تابع برگشتی زیر بر روی درخت دودوئی T چه عملی انجام می‌دهد؟**

پاسخ صحیح گزینه سوم یعنی تعداد کل گره‌های درخت را باز می‌گرداند. تعداد کل گره‌های درخت دودوئی برابر $۲^n-۱$ می‌باشد لذا این الگوریتم در حال محاسبه تعداد کل گره‌های درخت دودوئی می‌باشد.

پاسخ سوال ۷: **میانگین تعداد مقایسه‌های موفق در الگوریتم جستجوی دودوئی برای آرایه زیر کدام است؟**

$X_۰$	$X_۱$	$X_۲$	$X_۳$	$X_۴$	$X_۵$	$X_۶$	$X_۷$	$X_۸$
۱۰	۲۰	۴۰	۵۰	۷۰	۹۰	۱۱۰	۱۵۰	۲۰۰

پاسخ صحیح گزینه دوم یعنی $\frac{۲۵}{۹}$ است چون متوسط تعداد مقایسه‌ها برابر است با:

$$\frac{۱+۲\times۲+۴\times۳+۲\times۴}{۹}+\frac{۲۵}{۹}$$

چون برای رسیدن به پنجمین عنصر یک مقایسه و برای رسیدن به دومین و هفتمین عنصر دو مقایسه برای رسیدن به عنصرهای اول، سوم، ششم و هشتم، سه مقایسه و درنهایت برای رسیدن به عنصرهای چهارم و نهم، چهار مقایسه لازم است که در مجموع ۲۵ مقایسه لازم است پس میانگین آن می‌شود $\frac{۲۵}{۹}$.

پاسخ سوال ۹: **پیچیدگی زمانی الگوریتم مرتب‌سازی سریع ($QuickSort$) در بهترین حالت . . . . و در حالت متوسط . . . . و در بدترین حالت . . . . است:**

پاسخ صحیح گزینه اول می‌باشد: $T(n)=T(۰)+T(n-۱)+n-۱$

در بدترین حالت مرتب‌سازی سریع هنگامی است که آرایه از قبل مرتب باشد چون آرایه بطور مکرر به صورت یک زیر آرایه خالی در سمت چپ و یک زیر آرایه با $n-۱$ عنصر در سمت راست افراز می‌شود پس داریم: $T(n)=T(۰)+T(n-۱)+n-۱$

که در آن n زمان لازم برای افراز و $T(۰)$ زمان لازم برای مرتب‌سازی زیر آرایه سمت چپ به

طول صفر می‌باشد لذا $T(\circ)$ بوده و پیچیدگی زمانی در بدترین حالت برابر است با:

$$\begin{cases} T(n)=T(n-۱)+n-۱\ ,\ \ ,n>\circ \\ T(\circ)=\circ \end{cases}$$

لذا با روش تکرار و جایگذاری اثبات می‌شود: $T(n)=\frac{n(n-۱)}{۲}\in\theta(n^۲)$

پس مرتب‌سازی سریع در بدترین حالت از مرتبه $n^۲$ است و زمانی است که داده ها از قبل مرتب باشند.

مرتب‌سازی در حالت متوسط: $T(n)=\sum_{i=۱}^{k} s_ip_i$

$$T(n)=(n-۱)+\sum_{p=۱}^{n}\frac{۱}{n}[T(p-۱)+T(n-p)]$$

که $n-۱$ زمان لازم برای افراز است و $\frac{۱}{n}$ احتمال آنکه *pivot point* برابر p باشد و $T(p-۱)+T(n-p)$ نیز زمان میانگین برای مرتب‌سازی آرایه است و پس از ضرب در n وسط آن خواهیم داشت: $T(n)\leq ۲(n+۱)\ln n\in\theta(n\log n)\Longrightarrow T(n)\in\theta(n\log n)$

پس پیچیدگی زمانی در حالت میانگین و متوسط برای مرتب‌سازی‌های سریع و متوسط هر دو برابر $\theta(n\log n)$ هستند. البته در حالت متوسط آرایه تقریباً نصف شده و $QuickSort$ دوباره خودش را با نصف آرایه صدا می‌زند یعنی رابطه زیر را داریم:

$$T(n)+۲T(\tfrac{n}{۲})+\theta(n)\Longrightarrow T(n)=\theta(n\log n)$$

پاسخ سوال ۱۱: **تعداد مقایسه‌ها در الگوریتم بازگشتی پیدا کردن $\max$ و $\min$ عنصر در یک آرایه به روش تقسیم و غلبه در بدترین حالت کدام است؟**

پاسخ صحیح گزینه چهارم است: الگوریتم استاندارد برای یافتن $\max$ عنصر در یک آرایه S با n عنصر به صورت زیر است:

```
max = S[1];
   for(i = 2; i <= n; i++)
      if(S[i] > max)
         max = S[i];
```

$$T(n)=\begin{cases} \circ & ,\quad n=۱ \\ ۱ & ,\quad n=۲ \\ ۲T(\frac{n}{۲})+۲n & ,\quad n>۲ \end{cases}$$

رابطه فوق به روش تکرار حل می‌شود و خواهیم داشت:

$$T(n)=۲(۲T(\frac{n}{۴})+۲)+۲=۴T(\frac{n}{۴})+۶=۲^{k-۱}T(۲)+\sum_{i=۱}^{k-۱}۲^i=۲^{k-۱}+۲^k-۲=\frac{۳n}{۲}-۲$$

لذا تعداد مقایسه در بدترین حالت در الگوریتم بازگشتی پیدا کردن max و min برابر $\frac{۳n}{۲}-۲$ می‌باشد.

پاسخ سوال ۱۳: **پیچیدگی زمانی الگوریتم کروسکال در بدترین حالت کدام است؟**

پاسخ صحیح گزینه چهارم می‌باشد: هنگامی که گراف نسبتاً پر باشد و یال‌های زیادی داشته باشد تعداد یال‌ها برابر $\frac{n(n-۱)}{۲}$ می‌باشد یعنی $n-۱\leq e\leq\frac{n(n-۱)}{۲}$ که در آن e تعداد یال می‌باشد لذا:

$$T(e)=\theta(e\log e)\Longrightarrow\theta(e\log e)=\theta(n^۲\log n^۲)=\theta(n^۲.۲\log n)=\theta(n^۲\log n)$$

لذا اگر گراف یال‌های کمی داشته باشد بهتر است از الگوریتم پریم استفاده شود نه از الگوریتم کروسکال. الگوریتم کروسکال به‌صورت زیر تعریف می‌گردد:

$void\ kruskal(E,n,f)$

$\theta(\log m)\longleftarrow\{\ HEAP(E);$

$F=\phi$

$\theta(n)\longleftarrow initial(n);$

$\theta(m\log m)\longleftarrow while\ (number\ of\ edges<n-۱)$

$\{\ e=Delete(heap),$

$u-v=indices\ of\ vertices\ connected\ bye$

$i=find(u);$

$j=find(v);$

$if(i!=j)\ \{$

$merge(i,j);$

$f=F+\{e\};$

$\}$

$\}$

$\}$

پاسخ سوال ۱۵: **اشیاء زیر را در نظر بگیرید اگر ظرفیت کوله پشتی ۴۰ باشد جواب بهینه کوله پشتی با استفاده از روش حریصانه چیست؟**

X_i	X_1	X_2	X_3	X_4	X_5
P_i	۸	۵	۱۵	۱۰	۲۰
w_i	۱۶	۱۵	۲۵	۸	۱۵

پاسخ صحیح گزینه چهارم یعنی ۴۰٫۱ می‌باشد ابتدا نسبت $\frac{P_i}{w_i}$ را محاسبه می‌کنیم و اشیاء با بیشترین نسبت $\frac{P_i}{w_i}$ را در کوله پشتی قرار می‌دهیم:

$$\frac{P_i}{w_i} \quad \frac{8}{16}, \quad \frac{5}{15}, \quad \frac{15}{25}, \quad \frac{10}{8}, \quad \frac{20}{15}$$

اکنون به ترتیب نزولی مرتب می‌شود:

X_i	P_i	w_i
X_5	۲۰	۱۵
X_4	۱۰	۸
X_3	۱۵	۲۵
X_1	۸	۱۶
X_2	۵	۱۵

X_i	X_1	X_2	X_3	X_4	X_5
مرحله اول	۰	۰	۰	۰	۱
w	$=$ ۲۵				

X_i	X_1	X_2	X_3	X_4	X_5
مرحله دوم	۰	۰	۰	۱	۱
w	$=$ ۱۷				

X_i	X_1	X_2	X_3	X_4	X_5
مرحله سوم	۰	۰	$\frac{17}{25}$	۱	۱
w					

$$\sum_{i=1}^{5} P_i X_i = 8 \times 0 + 5 \times 0 + 15 \times \frac{17}{25} + 10 \times 1 + 20 \times 1 = 40{,}1$$

لذا جواب بهینه آن $x = (0, 0, \frac{17}{25}, 1, 1)$ خواهد بود.

پاسخ سوال ۱۷: **فرض کنید متنی شامل حروف a, b, c, d, e, f می‌باشد و تکرار آنها در متن جدول ذیل آمده است اگر تعداد کاراکترهای این متن ۸۹ باشد تعداد بیت‌های لازم برای نگهداری آن کدام است؟**

پاسخ صحیح گزینه ۱ یعنی ۱۷۷ بیت می‌باشد

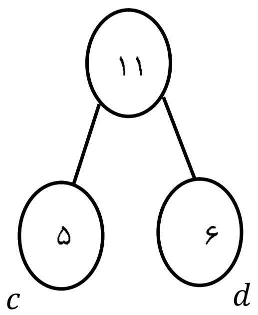
کاراکتر	a	b	c	d	e	f
تکرار	۲۵	۸	۵	٦	۳۵	۱۰

الگوریتم تولید کد هافمن را به کار می‌بریم:

مرحله اول:

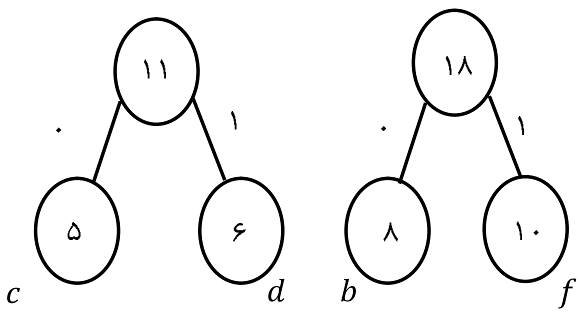
کاراکتر	c	d	b	f	a	e
به‌صورت صعودی	۵	٦	۸	۱۰	۲۵	۳۵

مرحله دوم: $a =$ ۲۵ و $e =$ ۳۵

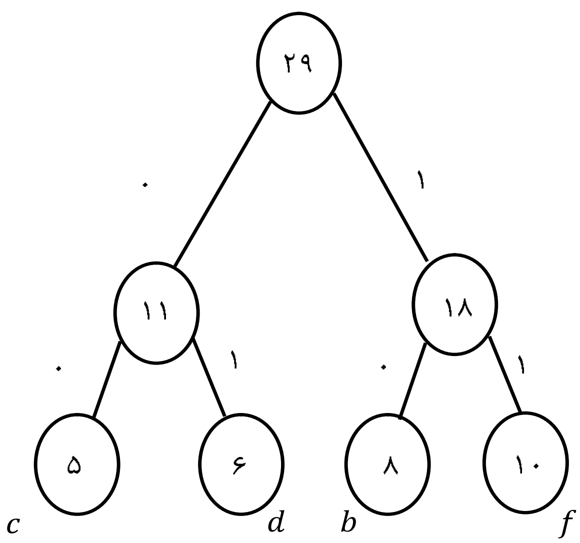


$b =$ ۸ و $f =$ ۱۰

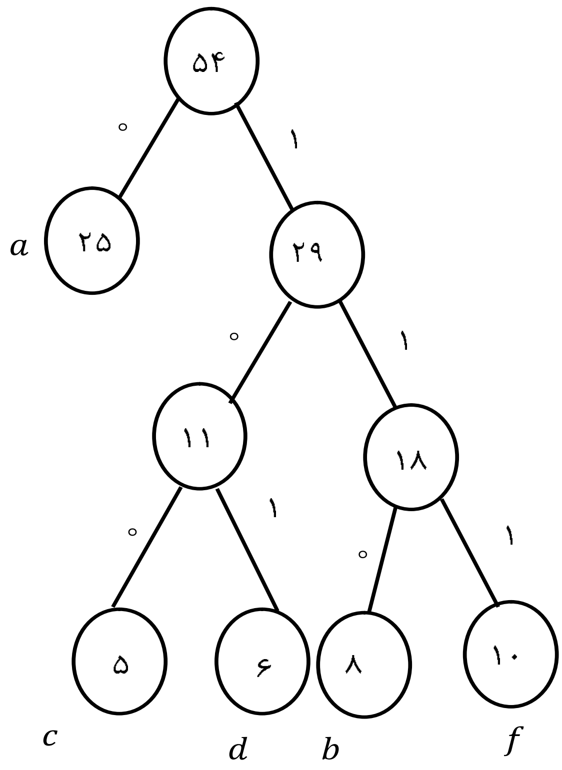
مرحله سوم: $a =$ ۲۵ و $e =$ ۳۵



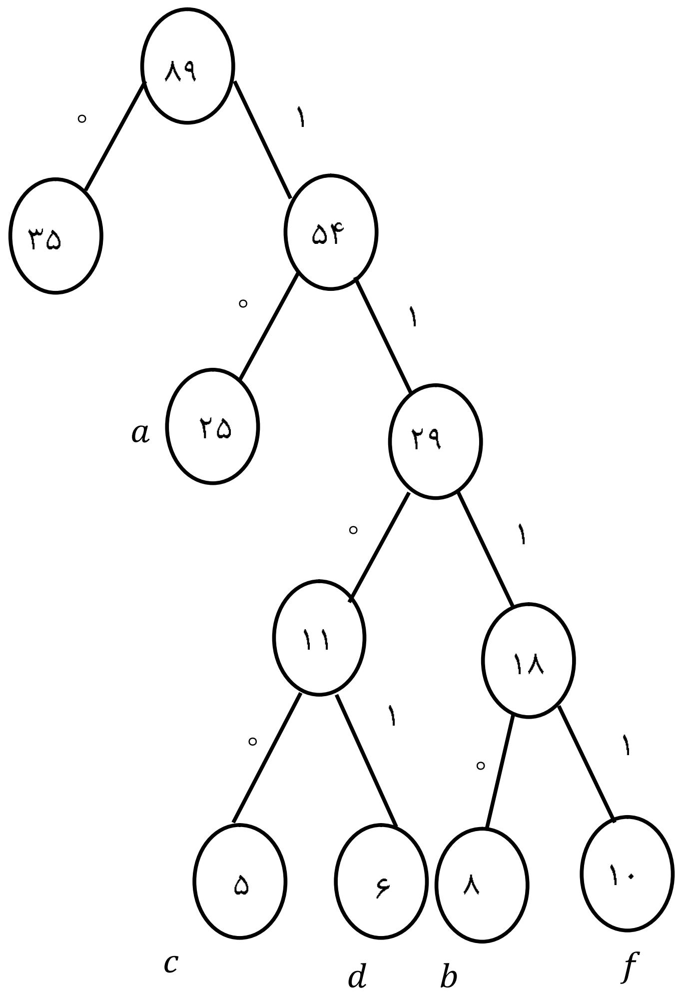
مرحله چهارم: $a =$ ۲۵ و $e =$ ۳۵



مرحله پنجم: $e = ۳۵$



مرحله آخر:



کاراکتر	a	b	c	d	e	f
کدها	۱۰	۱۱۱۰	۱۱۰۰	۱۱۰۱	۰	۱۱۱۱

تعداد کل بیت‌ها مجموع حاصل‌ضرب بیت‌های لازم برای هر کاراکتر در تعداد تکرار است:

$$۲ \times ۲۵ + ۹ \times ۸ + ۴ \times ۵ + ۱ \times ۳۵ + ۴ \times ۱۰ = ۱۷۷$$

پاسخ سوال ۱۹: **پیچیدگی زمانی الگوریتم فلوید در بدترین حالت کدام است؟**

پاسخ صحیح گزینه دوم یعنی $\theta(n^۳)$ می‌باشد چون الگوریتم فلوید برای کوتاه‌ترین مسیر به‌صورت زیر است:

$$\begin{cases} D = W \\ for(k = ۱; k <= n; k++) \end{cases}$$
$$\quad for(i = ۱; i <= n; i++)$$
$$\qquad for(j = ۱; j <= n; j++)$$
$$\qquad\quad D[i][j] = \min(D[i][j], D[i][k] + D[k][j]) \}$$

اندیس آرایه‌ها از ۱ تا n در نظر گرفته شده. چون از سه حلقه تو در تو تشکیل شده است. لذا مرتبه اجرایی آن $\theta(n^۳)$ می‌باشد.

$T(n) = n \times n \times n = n^۳$ پس مرتبه اجرایی الگوریتم فلوید $\theta(n^۳)$ می‌باشد.

در بدترین حالت نیز الگوریتم فلوید از همین پیچیدگی زمانی پیروی می‌کند.

پاسخ سوال ۲۱: **فرض کنید $T(n)$ تعداد روش‌های مختلف پارانتزگذاری حاصل‌ضرب n ماتریس باشد. آن‌گاه $T(n)$ کدام است؟**

پاسخ صحیح گزینه دوم یعنی $T(n) = \sum_{i=۱}^{n-۱} T(i)T(n-i)$ می‌باشد.

چون $M = M_۱ \times M_۲ \times M_۳ = (M_۱ \times M_۲) \times M_۳ = M_۱ \times (M_۲ \times M_۳)$

وقتی ماتریس‌ها زیاد باشند تعدا حالات شرکت‌پذیری ضرب آنها به شدت افزایش می‌یابد.

اگر $T(n)$ تعداد حالت‌های ممکن ضرب ماتریس‌های $M = (M_۱ \times M_۲ \times ... \times M_n)$ باشد.

خواهیم داشت: $T(n) = \sum_{i=۱}^{n-۱} T(i)T(n-i)$

$T(۱) = ۱ \Longrightarrow T(n)$ قابل محاسبه است.

مثال: $n = ۱۵ \Longrightarrow T(n) = ۲٦۷۴۴٦$

پاسخ سوال ۲۳: **مرتبه زمانی دقیق روال** $word\ series$ **برای محاسبه** $p(n, n)$ **کدام است؟**

پاسخ صحیح گزینه دوم می‌باشد یعنی $\theta(\frac{۴^n}{\sqrt{n}})$ می‌باشد.

پاسخ سوال ۲۵: **الگوریتم عقبگرد برای مسأله مدارهای همیلتونی دارای پیچیدگی زمانی . . . می‌باشد.**

پاسخ صحیح گزینه ۴ یعنی $\theta(n^n)$ می‌باشد. تابع الگوریتم عقبگرد برای مسأله مدارهای همیلتونی به صورت $void\ Hamilton(index\ i)$ می‌باشد.

```
{ index j
    if(promisiving(i))
    if(i == (n − 1))
        cout << vinde x[0] through vinde x[n − 1];
    else
        for(j = 2; j <= n; j + +)
            {vinde x[i + 1] = j;
            Hamilton(i + 1);
            }
    }
```

تعداد گره‌ها در درخت فضای حالت برای این الگوریتم عبارتست از:

$$۱ + (n - ۱) + (n - ۱)^۲ + ... + (n - ۱)^{n-۱} = \frac{(n - ۱)^n - ۱}{n - ۲}$$

که به‌وضوح مشخص است مرتبه زمانی الگوریتم از مرتبه $\theta(n^n)$ می‌باشد.

مسائل تشریحی ۹۵—۱۳۹۴
نیم‌سال اول

پاسخ سوال ۱: **رابطه بازگشتی زیر را به روش حدس و استقراء حل کنید.**

$$T(n) = T\left(\left\lceil \frac{n}{۲} \right\rceil\right) + T\left(\left\lfloor \frac{n}{۲} \right\rfloor\right) + ۱$$

فرض می‌کنیم $T(n) \leq C_n$ پس به استقراء

$$T(n) \leq C\left\lceil\frac{n}{۲}\right\rceil + C\left\lfloor\frac{n}{۲}\right\rfloor + ۱ = C_n + ۱ \geq C_n$$

اشتباه است لذا تصحیح می‌کنیم:

$$T(n) \leq C_n + B$$

فرض استقراء: $T(k) \leq C_k + B$ حکم استقراء: $T(n) \leq C_n + B$

داریم: $T(n) \leq T\left(\left\lceil\frac{n}{۲}\right\rceil\right) + T\left(\left\lfloor\frac{n}{۲}\right\rfloor\right) + ۱$

اثبات: برای $k = \frac{n}{۲}$ طبق فرض استقراء داریم: $T(n) = C_n + ۲B + ۱ \leq C_n + B \quad (۱)$

رابطه (۱) به‌ازای $B \leq -۱$ برقرار است پس

$$\forall n\ ;\ B \leq -۱ \Longrightarrow T(n) \leq C_n + B \Longrightarrow^{B=-۱} T(n) \leq C_n - ۱$$

پس $T(n) \in o(n)$

پاسخ سوال ۳: **نحوه محاسبه $\binom{۸}{۴}$ را با استفاده از برنامه‌نویسی پویا نشان دهید:**

	۰	۱	۲	۳	$۴ \leftarrow k$
۰	۱				
۱	۱	۱			
۲	۱	۲	۱		
۳	۱	۳	۳	۱	
۴	۱	۴	٦	۴	۱
۵	۱	۵	۱۰	۱۰	۵
٦	۱	٦	۱۵	۲۰	۱۵
۷	۱	۷	۲۱	۳۵	۳۵
$n \rightarrow ۸$	۱	۸	۲۸	۵٦	۷۰

سطر اول: $B[۰][۰] = ۱$

سطر دوم: $B[۱][۰] = ۱$

سطر دوم: $B[۱][۱] = ۱$

$B[۲][۰] = ۱$

$B[۲][۱] = B[۱][۰] \quad +B[۱][۱] = ۱ + ۱ = ۲$

سطر سوم: $B[۲][۲] = ۱$

پاسخ سوال ۵: **الگوریتم عقبگرد برای مسئله حاصل جمع زیرمجموعه‌ها را بنویسید.**

ورودی: عدد صحیح n، آرایه مرتب شده غیر نزولی w_i ها و عدد صحیح مثبت w.

خروجی: همه ترکیبات اعداد صحیح که حاصل جمع آنها مساوی w شود.

$$void\ sum\ of\ subsets(index\ i, int\ weight, int\ total)$$

$$\quad if(promising(i))$$

$$\qquad if(weight == w)$$

$$\qquad\quad cout << include[۱] throgh\ include[i];$$

$$\qquad else$$

$$\qquad\quad \{$$

$$\qquad\quad include[i+۱] ='' Yes'';$$

$$\qquad\quad sum\ of\ subsets(i+۱, w, weight + w[i+۱], total - w[i+۱]);$$

$$\qquad\quad include[i+۱] ='' No'';$$

$$\qquad\quad sum\ of\ subsets(i+۱, weight, total - w[i+۱]);$$

$$\qquad\quad \}$$

$$\qquad \}$$

$$\qquad boolpromising(index\ i);$$

$$\qquad\quad \{$$

$$\qquad return(weight + total \geq w)$$

$$\qquad\qquad\qquad (weight == w || weight + w[i+۱] <= w));$$

$$\qquad\quad \}$$

عنوان درس: طراحی الگوریتم	دانشگاه: پیام نور تهران شمال
استاد: آقای علی رضوی	پاسخ سوالات نیم‌سال اول ۹۸—۱۳۹۷
نام دانشجو: مژگان متقی	شماره دانشجویی: ۹۷۰۱۵۳۴۰۳

پاسخ سوال ۲: **اگر و فقط اگر ثابت C و عدد صحیح $n_\circ$ وجود داشته باشد که**

$\forall n \geq n_\circ;\ T(n) < Cg(n)$

پاسخ صحیح گزینه اول یعنی $T(n) \in O(g(n))$ می‌باشد چون طبق قضیه و تعریف داریم که اگر و فقط اگر ثابت C و ثابت صحیح $n_\circ$ وجود داشته باشد که برای همه مقادیر $n \geq n_\circ$ داشته باشیم: $T(n) < Cg(n)$، آن‌گاه $T(n) \in O(g(n))$ می‌باشد. این رابطه بصورت ریاضی و فرم زیر نیز قابل نمایش است:

$$T(n) \in O(g(n)) \Longleftrightarrow \exists c, n_\circ > \circ : \forall n \geq n_\circ;\ T(n) < Cg(n)$$

پاسخ سوال ۴: **مرتبه اجرائی رابطه بازگشتی زیر برابر کدام گزینه است؟**

پاسخ صحیح گزینه چهارم یعنی $\theta(n \log n)$ می‌باشد چون طبق روش اصلی و قضیه اصلی داریم. فرض کنید:

$$T(n) = aT(\tfrac{n}{b}) + F(n)\ ,\ a, b \geq ۱$$

حال برای محاسبه $T(n)$ داریم:

الف) اگر $F(n) \in O\left[n^{\log_b a - \varepsilon}\right]$ $(\varepsilon > \circ)$ دراین‌صورت $T(n) \in \theta\left[n^{\log_b a}\right]$

ب) اگر $F(n) \in \theta\left[n^{\log_b a}\right]$ دراین‌صورت $T(n) \in \theta\left[n^{\log_b a} \log_۲ n\right]$

ج) اگر $F(n) \in \Omega\left[n^{\log_b a + \varepsilon}\right]$ $(\varepsilon > \circ)$ دراین‌صورت $T(n) \in \theta\left[F(n)\right]$

البته چنانچه $aF(\frac{n}{b}) \leq CF(n)$ در این مسأله خواهیم داشت:

$$a = ۳, b = ۴\ ,\ f(n) = n \log n$$

$g(n)=n^{\log_b a}=n^{\log_۴ ۳}\in O(n^{۰/۷۹۳})$

$\varepsilon=۱\Longrightarrow f(n)\in\Omega\left[n^{\log_۳ ۹+\varepsilon}\right]\Longrightarrow$ درجه رشد $f(n)>$ درجه رشد $g(n)$

$T(n)\in\theta(n\log n)$

پاسخ سوال ٦: **یافتن بزرگترین عنصر در یک لیست مرتب از چه مرتبه زمانی است. (الگوریتم بهینه)**

پاسخ صحیح گزینه دوم می‌باشد چون چنانچه لیست مرتب باشد و به ترتیب سورت شده باشد یافتن بزرگترین عنصر از مرتبه $o(۱)$ بوده و تنها یک‌بار انجام می‌گردد. در غیر این‌صورت برای یافتن max باید $n-۱$ جستجو و مقایسه انجام گردد. طبق قضیه اثبات شده هر الگوریتمی که بتواند بزرگترین کلید در بین n کلید (برای هر ورودی دلخواه) فقط با مقایسه کلیدها بیابد باید در هر حالت حداقل $n-۱$ مقایسه انجام دهد ولی اگر آرایه مرتب شده باشد که در این سوال فرض بر مرتب بودن آن است بزرگترین کلید بدون هیچ مقایسه‌ای بدست می‌آید لذا $T(n)=o(۱)$ می‌باشد.

$\max=S[۱]$

$for(i=۲;i<=n,i++)$

$\quad if(S[i]>\max)$

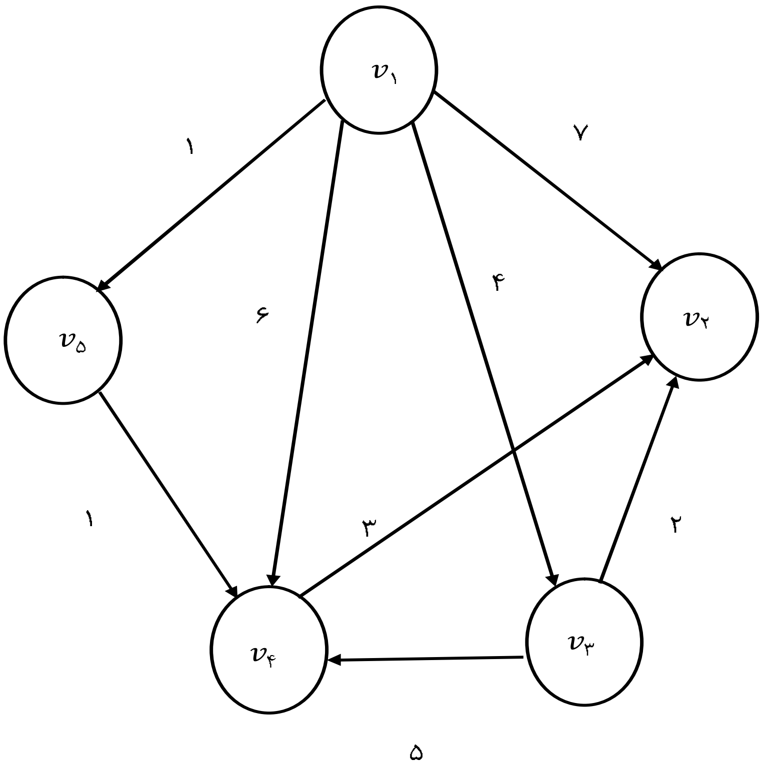
$\qquad \max=S[i];$

پاسخ سوال : ۸ خاصیت بهینه‌سازی زیرساختاری به چه معنی است؟

پاسخ صحیح گزینه اول می‌باشد به این معنی که یک مسأله دارای جواب بهینه است هرگاه زیرمسأله آن دارای جواب بهینه باشد.

پاسخ سوال ۱۰: **پیچیدگی زمانی الگوریتم دیکسترا از چه رتبه‌ای است؟**

پاسخ صحیح گزینه سوم یعنی $O(n^۲)$ می‌باشد برای محاسبه مرتبه زمانی الگوریتم دیکسترا باید نخست به حلقه $do-whil$ که $n-۱$ بار انجام می‌شود دقت کرد سپس متوجه می‌شویم که عملیات انجام شده متناسب با $\theta(n)$ می‌باشد لذا بوضوح مجموع زمان حلقه $O(n^۲)$ خواهد بود.



الگوریتم دیکسترا به‌صورت زیر می‌باشد:

$void\ Modify-Dikitra(v, cost, Dist, p, n)$

$\theta(n) \rightarrow for(i=0; i<n; i++)$

$\{$

$S[i]=0;$

$Disi[i]=cost[v][i];$

$P[i]='' v'';$

$\}$

$S[v]=1;$

do

$\{$

$\min=\infty;$

$for(i=1; i<n; i++)$

$if(0 \leq Dist[i] < \min)$

$$
\begin{aligned}
&\{\\
&\min = Dist + [i];\\
&u = i;\\
&\}\\
&S[u] = 1, P[u] = '' u'';\\
&for(w = 0; w < n; w++)\\
&if(Dist[w] > Dist[u] + cost[u][w]);\\
&\{\\
&Dist[w] = Dist[u] + cost[u][w];\\
&P[w] = P[u];\\
&\}\\
\theta(n-1) \rightarrow\ &\} while(n-1)time);\\
&\}
\end{aligned}
$$

$$
\begin{aligned}
&Y = \{v_1\}\\
&\underline{F = \phi}\\
&Y = \{v_1, v_5\}\\
&F = \{< v_1, v_5 >\}\\
&Y = \{v_1, v_5, v_4\}\\
&\underline{F = \{< v_1, v_5 >, < v_5 v_4 >\}}\\
&\vdots\\
&Y = \{v_1, v_5, v_4, v_3, v_2\}\\
&F = \{< v_1, v_5 >, < v_5, v_4 >, < v_1, v_3 >, < v_4, v_2 >\}
\end{aligned}
$$

از آنجا که در این الگوریتم در هر بازه فاصله گره با گره‌های قبلی مقایسه می‌شود الگوریتم از مرتبه زمانی $O(n^2)$ می‌باشد که n تعداد رئوس گراف می‌باشد.

پاسخ سوال ۱۲: **کمترین انتظار برای کارهای $P_1, P_2, ..., P_n$ زمانی حاصل می‌شود که:**

مرتبه زمان‌بندی الگوریتم فوق متناسب با مرتبه زمان‌بندی سورت آن $\theta(n \log n)$ می‌باشد پس پاسخ صحیح گزینه دوم می‌باشد چون طبق قضیه زمان کل در یک سیستم هنگامی کمینه می‌شود که کارها بر مبنای افزایش زمان ارائه خدمات مرتب شده و زمان‌بندی شوند از آنجا که مرتب‌سازی n قلم داده از مرتبه $\theta(n \log n)$ می‌باشد لذا داریم:

$\theta(n \log n) =$ مرتبه زمان‌بندی کمینه کردن زمان کل

الگوریتم زمان‌بندی:

$Sort(S, n)$

$for(i = \circ; i < n; i++)$

$\{$

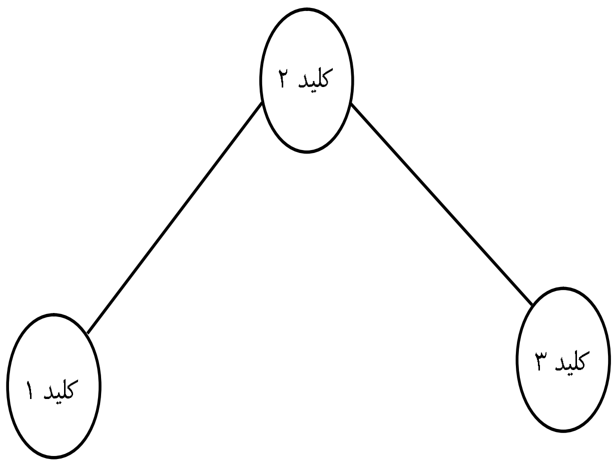
$X = selection(S);$

$F = F \cup \{x\};$

$\}$

پاسخ سوال ۱۴: **با داشتن احتمالات مربوط به جستجوی کلید مورد نظر در یک درخت دودوئی طبق جدول زیر زمان جستجوی میانگین برای درخت جستجوی زیر برابر است با:**

پاسخ صحیح گزینه دوم یعنی ۱/۸ می‌باشد چون $\sum_{i=1}^{n} c_i p_i =$ زمان میانگین جستجو

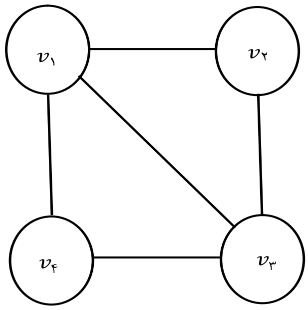


کلید	۱	۲	۳
احتمال	۰/۷	۰/۲	۰/۱

$$\sum_{i=1}^{n} c_i p_i = 2 \times 0/7 + 1 \times 0/2 + 2 \times 0/1 = 1/8$$

پاسخ سوال ۱٦: **در مسأله رنگ آمیزی گراف حداکثر انشعاب‌های خارج شده از هر گره برابر خواهد بود با:**

مسأله بهینه سازی رنگ‌پذیری به معنای یافتن حداقل مقدار m برای رنگ آمیزی گراف G است این عدد به عنوان عدد رنگی نامیده می‌شود.



پاسخ صحیح گزینه اول یعنی تعداد رنگ‌های داده شده برای رنگ آمیزی می‌باشد. چون در غیر این‌صورت رئوس مجاور هم‌رنگ می‌گردد بطور مثال

رئوس	v_1	v_2	v_3	v_4
رنگ	رنگ ۱	رنگ ۲	رنگ ۳	رنگ ۲

پاسخ سوال ۱۸: **مرتبه زمانی الگوریتم یافتن مدارهای هامیلتونی در یک گراف در بدترین شرایط برابر است با:**

پاسخ صحیح گزینه دوم یعنی $O(n^n)$ می‌باشد. با توجه به الگوریتم عقبگرد برای مدارهای هامیلتونی، تعداد گره‌ها در درخت فضای حالت برای این الگوریتم عبارتست از:

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

که به‌وضوح مشخص است مرتبه زمانی الگوریتم از درجه $\theta(n^n)$ می‌باشد.

پاسخ سوال ۲۰: **برای حل مسأله فروشنده دوره گرد به روش پویا با فرض داشتن یک گراف با ۴ رأس v_1, v_2, v_3, v_4 و با فرض شروع از رأس v_1 برای محاسبه $D[v_2][\{v_3, v_4\}]$ انتخاب از روی کدام موارد خواهد بود؟**

پاسخ صحیح گزینه دوم می‌باشد یعنی

$$D[v_2][\{v_3, v_4\}] = minmum \begin{cases} W[2][3] + D[v_3][\{v_4\}] \\ W[2][4] + D[v_4][\{v_3\}] \end{cases}$$

صحیح است. این محاسبه در گام سوم است که مجموعه A را بصورت دو عضوی در نظر می‌گیریم.

پاسخ سوال ۲۲: **در مورد الگوریتم هافمن کدام گزینه صحیح است؟**

پاسخ صحیح گزینه دوم می‌باشد یعنی حروف با تکرار بیشتر دارای کدهای با طول کمتر خواهند بود. در این روش کدگذاری از یک درخت دودوئی متشکل از تعدادی گره میانی و یال استفاده می‌کند که یال‌های درخت دارای وزن و بسته به موقعیت ارزش ۰ یا یک می‌گیرند.

پاسخ سوال ۲۴: **مسأله کوله پشتی صفر و یک و مسأله فروشنده دوره گرد در کدام دسته از مسائل دسته بندی می‌شوند؟**

پاسخ صحیح گزینه سوم می‌باشد. این‌گونه مسائل رام نشدنی بودن آنها ثابت نشده است ولی تاکنون هیچ الگوریتمی زمانی چندجمله‌ای برای آنها پیدا نشده است.

پاسخ سوالات تشریحی ۹۸–۱۳۹۷
نیم‌سال اول

پاسخ سوال ۲: **پیچیدگی زمانی الگوریتم** Quick Sort **را در هر دو حالت بدترین و حالت متوسط تحلیل نمائید.**

الف) در بدترین حالت: $T(n) = T(0) + T(n-1) + n - 1$

که در آن $T(0)$ زمان لازم برای مرتب‌سازی طرف چپ است، $T(n-1)$ زمان لازم برای مرتب‌سازی طرف راست است و $n-1$ زمان لازم برای تقسیم لیست است.

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

با روش تکرار و جایگزینی خواهیم داشت:

$$T(n) = T(n-1) + (n-1) = T(n-2) + (n-2) + (n-1) = \ldots = \frac{n(n-1)}{2}$$

$$\Longrightarrow T(n) \in \theta(n^2)$$

ب) در حالت متوسط: $T(n) = \sum\limits_{i=1}^{k} s_i p_i$

که s_i زمان لازم برای اجرای الگوریتم در حالت i ام می‌باشد و p_i احتمال وقوع.

$$T(n) = \frac{1}{n}\sum_{p=1}^{n}[T(p-1) + T(n-p)] + (n-1)$$

پس از ضرب در n و بسط:

$$nT(n) = n(n-1) + 2(T(0) + ... + T(n-1))$$

$$\vdots$$

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

$$a_n = \begin{cases} 0 & n < 1 \\ a_{n-1} + \dfrac{2(n-1)}{n(n+1)} & \end{cases} \qquad (1)$$

روابط فوق را می‌توان بصورت زیر نوشت:

$$a_n = a_{n-1} + \frac{2(n-1)}{n(n+1)} = a_{n-1} + 2\left(\frac{2n-1}{n} - \frac{2n}{n+1}\right)$$

با روش تکرار و جایگذاری خواهیم داشت:

$$a_n \leq 2\sum_{k=1}^{n}\frac{1}{k} = 2\int_1^n \frac{1}{x}dx = 2\ln n$$

پس

$$a_n \leq 2\ln n \Longrightarrow \frac{T(n)}{n+1} \leq 2\ln n \Longrightarrow T(n) \leq 2(n+1)\ln n$$

$$\Longrightarrow T(n) \in \theta(n\ln n)$$

پاسخ سوال ۴: **چهار ماتریس A, B, C, D را در نظر بگیرید**

$$A_{20\times 2} \times B_{2\times 30} \times C_{30\times 12} \times D_{12\times 8}$$

برای محاسبه حداقل تعداد ضرب در این زنجیره بروش برنامه نویسی پویا ماتریس m را تشکیل دهید و نسبت به تکمیل آن اقدام کنید.

بنا به خاصیت شرکت‌پذیری ضرب ماتریس‌ها می‌توان بصورت‌های مختلف ضرب $A \times B \times C \times D$ را انجام داد ولی حالت $A((BC)D)$ حالت بهینه است چون

$$A((BC)D) = 2 \times 30 \times 12 + 2 \times 12 \times 8 + 20 \times 2 \times 8 = 1232$$

$$m_{11} = m_{22} = m_{33} = m_{44} = 0$$

$$m_{12} = \min(m_{11} + m_{22} + 20 \times 2 \times 3) = 1200$$

$$m_{23} = \min(m_{22} + m_{33} + 2 \times 30 \times 12) = 720$$

$$m_{۳۴} = \min(m_{۳۳} + m_{۴۴} + ۳۰ \times ۱۲ \times ۸) = ۲۸۸۰$$

$$m_{۱۳} = \min(m_{۱۱} + m_{۲۳} + ۲۰ \times ۲ \times ۱۲,\ m_{۱۲} + m_{۳۳} + ۲۰ \times ۳۰ \times ۱۲) = ۱۲۰۰$$

$$m_{۲۴} = \min(m_{۲۲} + m_{۳۴} + ۲ \times ۳۰ \times ۸,\ m_{۲۳} + m_{۴۴} + ۲ \times ۱۲ \times ۸) = ۹۱۲$$

$$m_{۱۴} = \min(m_{۱۱} + m_{۲۴} + ۲۰ \times ۲ \times ۸,\ m_{۱۲} + m_{۳۴} + ۲۰ \times ۲ \times ۸,\ m_{۱۳} + m_{۴۴} + ۲۰ \times ۱۲ \times ۸ = ۱۲۳۲)$$

مرحله صفر	$\boxed{m_{۱۱} = ۰}$	$\boxed{m_{۲۲} = ۰}$	$\boxed{m_{۳۳} = ۰}$	$\boxed{m_{۴۴} = ۰}$
مرحله ۱	$\boxed{m_{۱۲} = ۱۲۰۰}$	$\boxed{m_{۲۳} = ۷۲۰}$	$\boxed{m_{۳۳} = ۲۸۸۰}$	
مرحله ۲	$\boxed{m_{۱۳} = ۱۲۰۰}$	$\boxed{m_{۲۴} = ۹۱۴}$		
مرحله ۳	$\boxed{m_{۱۴} = ۱۲۳۲}$			