سه‌شنبه

۱۸

10 October 2017 ۱۹ محرم ۱۴۳۹

مهر ۱۳۹۶

تمرین اول

۱ـ الف) الگوریتم پیدا کردن تعداد جابه‌جایی:

```
1. int T[n], i, j, counter;
2. for(i=0; i<n-1; i++){
3.     for(j=i+1; j<n-1; j++){
4.         if(T[i] > T[j])
5.             counter++;
6.     }
```

خط	هزینه	مرتبه تکرار	زمان
1	c_1	1	c_1
2	c_2	$n+1$	$c_2(n+1)$
3	c_3	$\sum_{i=0}^{n-1}(n-i)$	$c_3 \times \frac{n(n+1)}{2}$
4	c_4	$\sum_{i=0}^{n-1}(n-i-1)$	$c_4 \times \frac{n(n-1)}{2}$
5	c_5		

حالت اول (بهترین حالت): جابه‌جایی نداشته باشیم (سورت شده صعودی) $\leftarrow 0$

حالت دوم (بدترین حالت): سورت شده نزولی $\leftarrow \sum_{i=0}^{n-1}(n-i-1)$

بهترین حالت $= 0$

بدترین حالت $= c_5 \times \frac{n(n-1)}{2}$

در هر ۲ حالت (بهترین و بدترین): $T(n) = an^2 + bn + c$

یعنی در حالت متوسط هم: $T_{avg}(n) = a'n^2 + b'n + c$

★ در مقایسه با Insertion sort: در بهترین حالت Insertion sort تابع زمانی مرتبه اول $(T(n)=an+b)$ دارد اما در الگوریتم بالا تابع ما در بهترین حالت نیز درجه دو $(T(n)=an^2+bn+c)$ است. در بدترین و حالت متوسط رشد تابع زمان ۲ الگوریتم یکسان و درجه ۲ $(T(n)=an^2+bn+c)$ است.

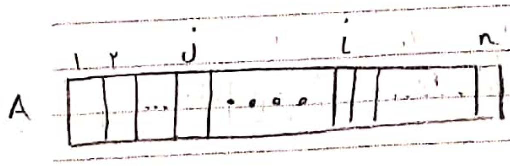
Scanned with CamScanner

★ نتیجه‌گیری: Ⓐ اگر تعداد نابه‌جایی‌ها صفر باشد به معنای آن است که آرایه ما از پیش به صورت sort شده و صعودی است. ← در این حالت طبق الگوریتم Insertion sort تابع زمانی ما رشد درجه اول خواهد داشت ($T(n)=an+b$)

Ⓑ اگر بدترین حالت در تعداد نابه‌جایی‌ها رخ بدهد (همانطور که در الگوریتم صفحه قبل گفته شد یعنی آرایه sort شده اما به صورت نزولی باشد) ← در این حالت تابع زمانی الگوریتم Insertion sort به صورت درجه دوم خواهد بود

($T(n)=an^2+bn+c$)

۱-ب) سعی می‌کنیم که آرایه اولیه انتخابی ما طوری باشد که کمترین نابه‌جایی را داشته باشد تا حداقل مقدار را بدست آوریم. در واقع کمترین حالت زمانی است که آرایه ما صعودی sort شده باشد و تنها جای ۲ خانه را باهم عوض کرده باشیم (چون خانه‌های دیگر در حالت sort شده با یکدیگر نابه‌جایی ایجاد نمی‌کنند.)



در آرایه sort شده و تعویض مکان داده i و j روبه‌رو
هیچ‌یک از خانه‌های $n, \ldots, i+1, i-1, \ldots, j+1, j-1, \ldots, 1$ باهم نابه‌جایی ندارند و فقط i و j آن را ایجاد کرده‌اند اما چون در حال حاضر خانه i از خانه‌های $i-1$ تا ... و $j+1$ ، j کوچکتر است و همچنین خانه j از خانه‌های $i-1$ تا ... و $j+1$ بزرگتر است پس به تعداد روبه‌رو حداقل نابه‌جایی داریم:
$(i-j)+(i-j)+1=2(i-j)+1$
خود i و j هم هستند

پنجشنبه
۲۰
12 October 2017 ۲۱ محـــرم ۱۴۳۹
مهر
۱۳۹۶

۲ـ کد را بازنویسی (مرتب‌تر) میکنیم:

```
1.  x = 0;
2.  for(i=1; i<n; i++){
3.      for(j=1; j<n; j++)
4.          x++;
4,1.    j=1;
5.      while(j<n){
6.          x++;  j=j*2;
7.      }
8.  }
```

خط	هزینه	مرتبه تکرار
1.	c_1	1
2	c_2	$n+1$
3	c_3	$n(n+1)$
4	c_4	$n\times(n)=n^2$
4,1	$c_{4,1}$	n
5	c_5	$n(n-1+1)=n^2$
6	c_6	$n(n-1)$
7	0	مهم نیست
8	0	مهم نیست

$T(n)=\sum$ (مرتبه تکرار)$\times$(هزینه)

$$T(n)=c_1+c_2(n+1)+c_3(n^2+n)+c_4(n^2)+nc_{4,1}+c_5n^2+c_6(n^2-n)$$

$$=\underbrace{(c_3+c_4+c_5+c_6)}_{a}n^2+\underbrace{(c_2+c_3-c_6+c_{4,1})}_{b}n+\underbrace{(c_1)}_{c}$$

$T(n)=an^2+bn+c \longrightarrow T(n)\in O(n^2)$ یا $T(n)\in\Theta(n^2)$

روز بزرگداشت حافظ

۳- Ⓐ در بهترین حالت : key در اولین خانه باشد.

$$T(n) = c_1 \times 1 + c_2\left(\frac{1}{10}\right) + c_3 = c_1 + \frac{c_2}{10} + c_3$$ عدد ثابت

	شبه کد	هزینه	تکرار در بهترین
1.	for(i=1; i<n; i++)	c_1	1
2.	if(A[i]==a)	c_2	$1 \times \frac{x}{10x} = 1/10$
3.	return i	c_3	1

Ⓑ در حالت میانگین :

خط	هزینه	تکرار
1	c'_1	$(n+1)$
2	c'_2	$\frac{1}{10} \times \left(1+2+\dots+\frac{n}{3}\right) + \frac{3}{10} \times \left(\left(\frac{n}{3}+1\right)+\left(\frac{n}{3}+2\right)+\dots+\left(\frac{2n}{3}\right)\right) + \frac{6}{10} \times \left(\left(\frac{2n}{3}+1\right)+\left(\frac{2n}{3}+2\right)+\dots+n\right)$
3	c'_3	1

$$T'(n) = (n+1)c'_1 + \frac{1}{10}\left(\frac{\left(1+\frac{n}{3}\right)\left(\frac{n}{3}\right)}{2}\right)c'_2 + \frac{3}{10}\left(\frac{(n+1)\left(\frac{n}{3}\right)}{2}\right)c'_2 + \frac{6}{10}\left(\frac{\left(\frac{5n}{3}+1\right)\left(\frac{n}{3}\right)}{2}\right)c'_2 + c'_3$$

$$= (n+1)c'_1 + \left(\frac{n}{60} + \frac{n^2}{180}\right)c'_2 + \left(\frac{n}{20} + \frac{n^2}{20}\right)c'_2 + \left(\frac{n}{10} + \frac{n^2}{6}\right)c'_2 + c'_3 \Rightarrow$$

$$T'(n) = \left(c'_1 + \frac{c'_2}{6}\right)n + \left(\frac{2}{9}c'_2\right)n^2 + c'_1 + c'_3$$ درجه ۲

ⓒ در بدترین حالت: در آخرین خانه باشد

خط	هزینه	تکرار
1	c_1''	$n+1$
2	c_2''	$0\times(1+2+\cdots+\frac{n}{3}) + 0\times((\frac{n}{3}+1)+(\frac{n}{3}+2)+\cdots+(\frac{2n}{3})) + \frac{6}{10}\times((\frac{2n}{3}+1)+(\frac{2n}{3}+2)+\cdots+n)$
3	c_3''	1

$$T''(n) = (n+1)c_1'' + \left(\frac{n}{10}+\frac{n^2}{6}\right)c_2'' + c_3''$$

$$= \left(\frac{c_2''}{6}\right)n^2 + \left(c_1'' + \frac{c_2''}{10}\right)n + c_1'' + c_3''$$ درجه ۲

A

۴- اگر آرایه مرتب شده باشد : (بهترین حالت را در نظر می‌گیریم از این به بعد)

Arr [1 . . . . . n]

```
1. for( i=1 ; i<n-1 ; i++){
2.     for( j=n ; j>i ; j--){
3.         if( Arr[i]+Arr[j] < k)
4.             break;   (به خط شماره ۵ برمی‌گردد)
5.         if( Arr[i]+Arr[j] == k)
6.             return true;
7.     }
8. }
```

خط	هزینه	تعداد تکرار
1	c_1	n
2	c_2	$\sum_{i=1}^{n-1}(n-i+1) = \frac{(n-1)(n+2)}{2}$
3	c_3	$\sum_{i=1}^{n-1}(n-i) = \frac{n(n-1)}{2}$
4	c_4	بهترین حالت = ۰ بدترین حالت = $\sum_{i=1}^{n-1}(n-i-1) = \frac{(n-2)(n-1)}{2}$
5	c_5	$\sum_{i=1}^{n-1}(n-i) = \frac{n(n-1)}{2}$
6	c_6	1 $\Longrightarrow$ $T(n) = an^2+bn+c$
7, 8	۰	مهم نیست

(B) اگر آرایه مرتب نشده باشد: (بدترین حالت را در نظر میگیریم)

```
1. for(i=1 ; i<n; i++){
2.     for(j=i+1; j<=n; j++){
3.         if(Arr[i]+Arr[j]==k)
4.             return true;
5.     }
6. }
```

خط	هزینه	تعداد تکرار
1	c_1	n
2	c_2	$\sum_{i=1}^{n-1}(n-i+1) = \frac{(n+2)(n-1)}{2}$
3	c_3	$\sum_{i=1}^{n-1}(n-i) = \frac{n(n-1)}{2}$
4	c_4	1
5	۰	مهم نیست
6	۰	مهم نیست

زمان الگوریتم $\Longrightarrow T(n) = an^2+bn+c$

هر دو الگوریتم در زمان $O(n^2)$ انجام شدند.

ه

a) $\lg(n!) = \Theta(\lg(n^n))$

if $A = O(B)$ & $A = \Omega(B) \implies A = \Theta(B)$

اول: $A = O(B) \longleftrightarrow \exists c > 0 , n_0 > 0 \mid \forall n \geq n_0 \quad 0 \leq A \leq c.B$

از استقرا استفاده میکنیم : پایه $\longrightarrow c = 1 , n_0 = 1 \quad \lg(1) \leq \lg(1)$ ✓

فرض استقرا $\longrightarrow \lg(n!) \leq \lg(n^n)$

مرحله استقرا $\longrightarrow \lg(n!) \leq \lg(n^n) \longrightarrow \lg(n+1) + \lg(n!) \leq \lg(n+1) + \lg(n^n)$

$\longrightarrow \lg((n+1)!) \leq \lg(n+1) + n\lg(n)$ ①

چون تابع لوگاریتم اکیداً صعودی است (از $n \geq 1$) : $\lg(n) \leq \lg(n+1)$ ②

①,② $\longrightarrow \lg((n+1)!) \leq \lg(n+1) + n\lg(n+1) = (n+1)\lg(n+1) = \lg((n+1)^{n+1})$

$\longrightarrow \lg((n+1)!) \leq \lg((n+1)^{(n+1)})$ حکم ثابت شد ✓

دوم : $A = \Omega(B) \longleftrightarrow \exists c > 0 , n_0 > 0 \mid \forall n \geq n_0 \quad 0 \leq c.B \leq A$

در واقع برای یافتن c میتوان : $\lg(n!) \geq c.\lg(n^n) \longrightarrow c \leq \underbrace{\frac{\lg(n)}{n\lg(n)} + \frac{\lg(n-1)}{n\lg(n)} + \dots \frac{\lg(1)}{n\lg(n)}}_{A \text{ عبارت}}$

$\longrightarrow c \leq A \leq n \times (\frac{1}{n}) \longrightarrow \boxed{c \leq 1}$

یعنی اگر $c \leq 1$ باشد و $n_0 = 1$) چون : $\lg(1) \leq \lg(1)$ پایه برقرار است ✓)

حکم ثابت می‌شود.

b) اثبات مستقیم : $\lim_{n\to\infty} \frac{n^{1/\lg n}}{1} \longrightarrow$ ابتدا داریم $\lim_{n\to\infty} \frac{1}{\lg n} = 0$

$\longrightarrow \lim_{\substack{n\to\infty \\ \alpha\to 0}} \frac{n^{\alpha}}{1} = 1$ ✓ حکم ثابت شد

روش دوم (دو مرحله‌ای) : $\exists c>0, n_0>0 \mid \forall n\geq n_0 \quad 0\leq c_1\cdot g(n)\leq f(n)\leq c_2\cdot g(n)$

مرحله ۱: $c_1\cdot 1\leq n^{1/\lg(n)} \longrightarrow \boxed{c_1=1 \,,\, n_0=2} \quad 1\leq 2^{1/\lg 2}=2$ ✓

مرحله ۲: $n^{1/\lg n}\leq c_2\cdot 1 \xrightarrow{\lg} \lg(n^{1/\lg n})\leq \lg c_2 \longrightarrow \frac{\lg(n)}{\lg(n)}\leq \lg c_2 \longrightarrow \boxed{c_2\geq 2}$ ✓

✓ $2\geq 2^{1/\lg 2}=2$ یعنی اگر c_2 را مثلاً ۲ انتخاب کنیم و $\boxed{n_0=2}$ معادله برقرار است

پس ↓

Ⓐ و Ⓑ $\longrightarrow \left(n^{\frac{1}{\lg n}}=\Theta(1)\right) \,\&\, \left(n^{\frac{1}{\lg n}}=\Omega(1)\right) \Longrightarrow n^{\frac{1}{\lg n}}=\Theta(1)$

c) $n! = \omega(2^n) \longrightarrow \lim_{n\to\infty} \frac{\sqrt{2\pi n}\left(\frac{n}{e}\right)^n\left(1+\Theta\left(\frac{1}{n}\right)\right)}{2^n}$ $\Rightarrow$ توان های بزرگ هر جمله را نگه میداریم و اعداد ثابت را حذف میکنیم

$\lim_{n\to\infty} \frac{(n^{\frac{1}{2}})\times(n^n)\times(n^{-1})}{2^n\times e^n} = \lim_{n\to\infty} \frac{n^{n-\frac{1}{2}}}{(2e)^n} = \lim_{n\to\infty} \frac{n^n}{k^n} = \lim_{n\to\infty} \left(\frac{n}{k}\right)^n$

هم پایه و هم توان به سمت بی نهایت میروند پس حد برابر ∞ است

$\lim_{n\to\infty} \left(\frac{n}{k}\right)^n = \infty \longrightarrow n!$ رشد $> 2^n$ رشد ✓

روش دوم صفحه بعد

روش دوم (بدون استرلینگ) :

$\forall c>0 \,,\, \exists n_0>0 \mid \forall n \geq n_0$ داریم $c.g(n) < f(n)$

حال میخواهیم بگوییم به ازای هر c که گرفته شود میتوان یک n_0 نام برد که شرط مساله را برآورده کند.

$n! = \omega(2^n)$?

$2^n \times c < n! \rightarrow c \times \underbrace{\overbrace{\frac{2}{n-1} \times \dots \times \frac{2}{3} \times \frac{2}{2} \times \frac{2}{1}}^{\text{عبارت } A}}_{<1} < n$

$\rightarrow 2c < A \text{ عبارت } < n \rightarrow 2c < n \rightarrow n_0 = \lceil 2c \rceil + 1$

اگر فلش ها را از آخر به اول برعکس کنیم به اثبات دلخواه میرسیم.

d) $n! = o(n^n) \rightarrow \lim\limits_{n\to\infty} \frac{\sqrt{2\pi n}\left(\frac{n}{e}\right)^n\left(1+\Theta\left(\frac{1}{n}\right)\right)}{n^n} = 0$ باید اثبات شود

$\rightarrow \lim\limits_{n\to\infty} \frac{\sqrt{2\pi n}\left(1+\Theta\left(\frac{1}{n}\right)\right)}{e^n} = \lim\limits_{n\to\infty} \frac{(n^{\frac{1}{2}})(n^{-1})}{e^n} = \lim\limits_{n\to\infty} \frac{1}{\sqrt{n}\, e^n} = 0$ ✓

۶-

a) غلط : $n^3 \in O(n^4)$ اما $n^4 \notin O(n^3)$

b) غلط : $\sqrt{n} + n \notin \Theta(\sqrt{n})$

c) درست : $f(n) = O(g(n))$ یعنی $f(n) \leq c.g(n) \xrightarrow{\text{لگاریتم میگیریم}}$

$\lg(f(n)) \leq \lg(c.g(n)) \rightarrow \lg(f(n)) \leq \lg(c) + \lg(g(n))$ ①

$\lg(c) + \lg(g(n)) \leq (\lg(c)+1)\lg(g(n))$ ② ✓

①,② $\rightarrow \lg(f(n)) \leq \underbrace{\cancel{(\lg(c)+1)}}_{\text{ثابت } k} \lg(g(n)) \rightarrow \boxed{\lg(f(n)) = O(\lg(g(n)))}$

d) غلط: $2n = \Theta(n)$ اما $2^{2n} \neq \Theta(2^n)$

e) غلط: $\frac{1}{n} \neq \Theta(\frac{1}{n^2})$

f) درست: $f(n) = \Theta(g(n))$ یعنی $\exists c>0, n_0>0 \mid \forall n \geq n_0 \quad 0 \leq f(n) \leq c\cdot g(n)$

$\longrightarrow 0 \leq \frac{1}{c} f(n) \leq g(n) \longrightarrow 0 \leq k f(n) \leq g(n)$ ①

میدانیم $c>0$ آنگاه مثبت $k=\frac{1}{c}$ و به ازای هر c مثبت یک k وجود دارد.

① $\xrightarrow{\text{بازنویسی}}$ $\exists c>0, n_0>0 \mid \forall n \geq n_0 \quad 0 \leq k f(n) \leq g(n) \iff g(n) = \Omega(f(n))$

با بازنویسی عبارت اول به نتیجه میرسیم.

g) غلط: $f(n) = 2^{4n}$ اگر $\longrightarrow f(\frac{n}{2}) = 2^{2n} \Rightarrow 2^{4n} \neq \Theta(2^{2n})$

h) درست: if $f(n) \geq 0$ then $o(f(n)) \geq 0 \longrightarrow f(n) + o(f(n)) \geq f(n)$

$\Longrightarrow f(n) + o(f(n)) = \Omega(f(n))$ ①

② $f(n) \geq g(n) = o(f(n))$ n_0 را طوری میگیریم که برای $n \geq n_0$ است آنگاه

$\longrightarrow f(n) + o(f(n)) \leq 2f(n) \longrightarrow f(n) + o(f(n)) = \Theta(2f(n)) = \Theta(f(n))$

①,② $\longrightarrow$ ثابت شد ✓

پنجشنبه
۱۳
5 October 2017 ۱۴۳۹ محرم ۱۴
مهر
۱۳۹۶

$A = \ldots(B)$

A	B	O	o	Ω	ω	Θ
n^2	n^3	✓	✓	×	×	×
$\lg^k n$	n^{ϵ}	✓	✓	×	×	×
n^k	c^n	✓	✓	×	×	×
2^n	$2^{n/2}$	×	×	✓	✓	×
$n^{\lg c}$	$c^{\log n}$	✓	×	✓	×	✓
$4^{\lg n}$	n^2	✓	×	✓	×	✓
$n!$	$n \times 2^n$	×	×	✓	✓	×
$\sqrt{2}^{\lg n}$	$2^{\sqrt{2 \cdot \lg(n)}}$	×	×	✓	✓	×
$(\lg(n)!)$	2^{2^n}	✓	✓	×	×	×
$n^{\lg(\lg(n))}$	$(\lg(n))^{\lg(n)}$	✓	×	✓	×	✓

α

α

$$2^{\lg(\sqrt{n})} \overset{?}{=} o\left(2^{\sqrt{\lg(n^2)}}\right)$$

$$\lg(\sqrt{n}) \overset{?}{=} o\left(\sqrt{\lg(n^2)}\right) \longleftarrow \lg(\sqrt{n}) < c \cdot \sqrt{\lg(n^2)} \longleftarrow \lg^2\sqrt{n} < K \cdot \lg(n^2)$$

ثابت ثابت

$$\longleftarrow \lg(n^{1/2}) \times \lg(n^{1/2}) = \cancel{\tfrac{1}{4}} \lg^2(n) < \cancel{2K} \lg(n) \longleftarrow \lg^2(n) < \lg(n) \quad \times$$

یعنی: $2^{\lg(\sqrt{n})} = \omega\left(2^{\sqrt{\lg(n^2)}}\right)$