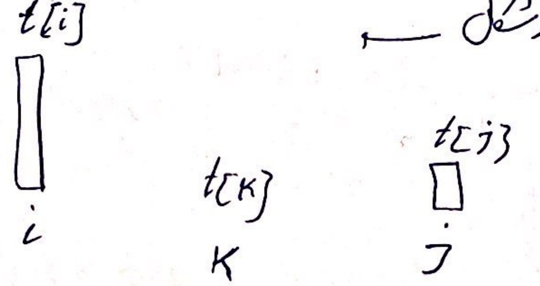
فرشید نوش   ۹۸۳۱۰۶۸

۱-

الف) ادعا می‌کنم که زمان اجرای الگوریتم کندک‌تر سازی تعداد نابجایی هاست فرض کنید آرایه‌ی اولیه k تا نابجایی داشته باشد، هر عمل مقایسه و swap در عملیات insertionSort دقیقاً یک واحد از این تعداد نابجایی ها کم می‌کند (اگر swap که انجام شود) (دو عنصری جابجا شده با هم نابجایی داشته اند نابجایی باقی عناصر هم با swap دو عنصر متوالی تغییر نمی‌کند) پس زمان مورد نیاز برای اجرای این الگوریتم حداقل برابر تعداد نابجایی‌های آرایه است.

ب) هر عنصر میان i و j را در نظر بگیرید مانند شکل ←



$t[K] > t[j]$ ⟶ K با j نابجایی دارند ⟶ هر عنصر این بین حداقل یک نابجایی دارد

$t[K] < t[j]$ ⟶ K با i نابجایی دارند

حداقل تعداد نابجایی عناصر = $j-i$

⟵ حداقل تعداد نابجایی ها = $\underbrace{(j-i-1)}_{\text{عناصر میانی}} + \underbrace{1}_{\text{نابجایی i و j}}$

۲-

	code	cost	times
1.	`x=0`	c_1	1
2.	`for(i=1; i<=n; i++){`	c_2	n
3.	`for(j=1; j<=n; j++)` `x++;`	c_3	$\sum_{i=1}^{n} n = n^2$
4.	`j=1;`	c_4	$\sum_{i=1}^{n} n = n^2$
5.	`while(j<n){`	c_5	n
6.	`x++; j=j*2;`	c_6	$\lceil \log(n) \rceil + 1$ ← دفعه‌ی آخر از حلقه بیرون می‌رود
7.	`}`	c_7	$\lceil \log(n) \rceil$
8.	`}`		

فرشید نوش ۹۸۳۱۰۶۸

۳-

```
for (int i=1; i<=n; i++)    c1 × 1
    if(A[i]==a)             c2 × 1
        return i;           c3 × 1
```

⟶ $c_1 + c_2 + c_3 \rightarrow O(1)$

بهترین حالت : a عنصر اول آرایه باشد ⟵

یک بار وارد حلقه ی فور می شود و یک بار شرط چک می شود و همان مرتبه هم return می شود.

```
for (int i=1; i<=n; i++)    c1 × n
    if (a[i]==a)            c2 × n
        return i;           c3 × 1
```

⟶ $n\,c_1 + n\,c_2 + c_3 \rightarrow O(n)$

بدترین حالت : a عنصر آخر آرایه باشد ⟵

حلقه n بار تکرار شده و در هر بار هم if چک می شود و تنها در مرحله ی آخر وارد بدنه ی if می شود.

حالت متوسط:

$$O\left(\sum_{i=1}^{n/3} x\, i + \sum_{i=n/3}^{2n/3} 3x\, i + \sum_{i=2n/3}^{n} 6x\, i\right) =$$

$$O\left(\frac{x\frac{n}{3}\left(\frac{n+3}{3}\right)}{2} + 3x\left(\frac{\frac{2n}{3}\cdot\frac{2n+3}{3}}{2} - \frac{\frac{n}{3}\left(\frac{n-3}{3}\right)}{2}\right) + 6x\left(\frac{n(n+1)}{2} - \frac{\frac{2n}{3}\left(\frac{2n-3}{3}\right)}{2}\right)\right)$$

$$= O\left(x\, n^2 c_1 + 3x\, n^2 c_2 + 6x\, n^2 c_3\right) = O(xn^2) \simeq O(n)$$

$$\boxed{\frac{n}{3}x + \frac{n}{3}3x + \frac{n}{3}6x = 1 \rightarrow x = \frac{3}{10n}}$$

$$x = \frac{3}{10n} \rightarrow O(xn^2) = O(cn) = O(n)$$

فرشید نوش

۴-

1- نامرتب

	Const	times
`for (int i=1; i<=n; i++)` →	c_1	n
`for (int j=1; j<=n; j++)` →	c_2	$\sum_{i=1}^{n} n = n^2$
`if (a[i]+a[j]==K)` →	c_3	$\sum_{i=1}^{n} n = n^2$
`return true;` →	c_4	حداکثر یکبار

```
for (int i=1; i<=n; i++)
    for (int j=1; j<=n; j++)
        if (a[i]+a[j]==K)
            return true;
return false;
```

$$\rightarrow O(c_1 n + c_2 n^2 + c_3 n^2 + c_4) = \boxed{O(n^2)}$$

2- مرتب

```
int j = n;
for (int i=1; i<=n; i++) {
    while (j>=1 && a[i]+a[j] > K)
        j--;
    if (a[i]+a[j]==K)
        return true;
}
return false;
```

طبق الگوریتم two pointers در اینجا عمل کردم که چون در اینجا از پوینتر دوم حداکثر n بار کم می‌شود و به پوینتر اول حداکثر n بار اضافه می‌شود پیچیدگی زمانی ما از $O(n)$ هست. البته چون آرایه sort شده بود توانستم از این الگوریتم استفاده کنیم.

۹۸۳۱۰۴۸ فرشید نوشی

۵-

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

$\lg(n!) = \sum_{i=1}^{n} \lg(i) \leq n \lg n = \cancel{\Theta(n \lg n)} \rightarrow \boxed{\lg(n!) = O(n \lg n)}$

$\lg(n!) = \sum_{i=1}^{n} \lg(i) \geq \sum_{i=\frac{n}{2}}^{n} \lg i \geq \frac{n}{2} \lg \frac{n}{2} \rightarrow \lg(n!) = \Omega\left(\frac{n}{2} \lg \frac{n}{2}\right)$

$\Omega\left(\frac{n}{2} \lg \frac{n}{2}\right) = \Omega\left(n \lg\left(\frac{n}{2}\right)\right) = \Omega\left(n(\lg n - \lg 2)\right) = \boxed{\Omega(n \lg n)}$

b- $n^{\frac{1}{\lg n}} = \Theta(1)$

$n^{\frac{1}{\lg n}} = n^{\lg_n^2} = 2 \rightarrow n^{\frac{1}{\lg n}} = 2 = \Theta(1)$

c- $n! = \omega(2^n)$

$f(n) = \omega(g(n)) \Longleftrightarrow \lim_{n \to \infty} \frac{g(n)}{f(n)} = 0$

$\lim_{n \to \infty} \frac{2^n}{\left(\frac{n}{e}\right)^n \sqrt{2\pi n}\left(1+\Theta\left(\frac{1}{n}\right)\right)} = \lim_{n \to \infty} \frac{2^n e^n}{\left(1+\Theta\left(\frac{1}{n}\right)\right) n^n \sqrt{2\pi n}} =$

$\lim_{n \to \infty} \frac{1}{\left(1+\Theta\left(\frac{1}{n}\right)\right)\left(\frac{n}{2e}\right)^n \sqrt{2\pi n}} = 0$

d- $n! = o(n^n)$

$f_n = o(g(n)) \Longleftrightarrow \lim_{n \to \infty} \frac{g(n)}{f(n)} = \infty$

$\lim_{n \to \infty} \frac{n^n}{n!} = \lim_{n \to \infty} \frac{n^n}{\left(1+\Theta\left(\frac{1}{n}\right)\right)\left(\frac{n}{e}\right)^n \sqrt{2\pi n}} = \lim_{n \to \infty} \frac{e^n}{\left(1+\Theta\left(\frac{1}{n}\right)\right)\sqrt{2\pi n}} = \infty$

فرشید نوش 9831048

4.

a) $f(n) = n,\ g(n) = n^2$ ← مثال نقض غلط $n^2 = O(n)$ ✗

b) $f(n) = n,\ g(n) = n^2 \longrightarrow n + n^2 = \Theta(n^2)$ ✗

c) $f(n) = O(g(n)) \longrightarrow \exists\, n_0, c,\ \forall n > n_0\ \ f(n) < c\,g(n)$

$\xrightarrow{n_0, c} \lg(f(n)) < \lg(c\,g(n)) = \lg(g(n)) + \lg(c)$

$\underbrace{N_0 = n_0,\ C = c}$ $\longrightarrow \lg(f(n)) = O(\lg(g(n)))$

$\forall n > N_0 \rightarrow \lg(f_n) < c \lg(g(n))$

فرض $\lg(f(n)) < \lg(c\,g(n)) = \lg(g(n)) + \lg(c) < c\lg(g(n))$

d) $f(n) = 3n,\ g(n) = n$

$2^{3n} = 8^n > 2^n$

$\rightarrow 2^{f(n)} = O(2^{g(n)})$ ✗

e) $f(n) = \frac{1}{n} \longrightarrow f(n) = O(f(n^2)) \longrightarrow$

$\frac{1}{n} = O(\frac{1}{n^2})$ ✗

$\forall n > 1\ \ (\frac{1}{n^2} < \frac{1}{n})$

f) $f(n) = O(g(n)) \longrightarrow g(n) = \Omega(f(n))$

$\exists\, n_0, c \rightarrow \forall n > n_0\ \ f(n) < c\,g(n) \longrightarrow$

$N_0 = n_0,\ C = c \rightarrow \forall_{n > n_0}\ f(n) < c\,g(n)$ ☆ فرض

g) $f(n) = 3^n,\ f(n/2) = 3^{n/2} \longrightarrow \exists\, c_1, c_2, n_0 > 0\ \{c_1 3^{n/2} < 3^n < c_2 3^{n/2}\}$

$\longrightarrow c_1 < 3^{n/2} < c_2$ ✗

برای هر n بزرگی این رابطه برقرار نیست.

فرشید نوش

h) $f(n) + o(f(n)) = \Theta(f(n))$

لم ۱) $f(n) + g(n) = \Theta(\max(f(n), g(n)))$

$\max(f(n), g(n)) = h(n)$

$f(n) \leqslant h(n)$
$g(n) \leqslant h(n)$ $\rightarrow$ $f(n) + g(n) \leqslant 2h(n) \longrightarrow f(n) + g(n) = O(h(n))$

& $\rightarrow$ $f(n) + g(n) \geqslant h(n) \longrightarrow f(n) + g(n) = \Omega(h(n))$

$\rightarrow f(n) + g(n) = \Theta(h(n))$

طبق لم ۱ $\rightarrow f(n) + o(f(n)) = \Theta(\max(f(n), o(f(n)))) = \Theta(f(n))$

$(\max(f(n), o(f(n))) = f(n))$

7-

A	B	O	o	Ω	ω	Θ
n^2	n^3	yes	yes	no	no	no
$lg^k n$	n^{ε}	yes	yes	no	no	no
n^k	c^n	yes	yes	no	no	no
2^n	$2^{n/2}$	no	no	yes	yes	no
$n^{lg\, c}$	$c^{log\, n}$	no	no	yes	yes	no
$4^{lg\, n}$	n^2	yes	no	yes	no	yes
$n!$	$n.2^n$	no	no	yes	yes	no
$\sqrt{2}^{lg\, n}$	$2^{\sqrt{2.lg(n)}}$	no	no	yes	yes	no
$(lg(n))!$	2^{2^n}	yes	yes	no	no	no
$n^{lg(lg(n))}$	$(lg(n))^{lg(n)}$	yes	no	yes	no	yes