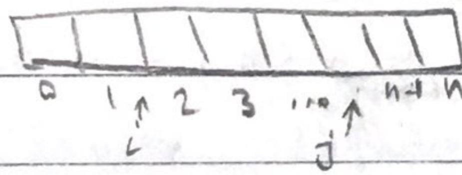

	cost	
For $j \leftarrow 2$ to n	C_0	n
do key $\leftarrow A[j]$	C_1	$n-1$
$i \leftarrow j-1$	C_2	$n-1$
while $i > 0$ and $A[i] >$ key	C_3	$\sum_{j=2}^{n} t_j$
do $A[i+1] \leftarrow A[i]$	C_4	$\sum_{j=2}^{n} (t_j - 1)$
i ~~←~~	C_5	" "
$A[i+1] \leftarrow$ key	C_6	$n-1$

الف - چون سوال گفته که فقط یک نابجایی وجود دارد تا زمانی که به for بیرونی به اولین نابجایی نرسیده ایم وارد while نمی شود. با رسیدن به i هم وارد while نمی شود اما با رسیدن به $i+1$ می بینیم که باید یک خونه رو جابجا کند ...

برای مثال

1	2	6	4	5	3	7	8
		i			j		

۱- در ابتدا این خانه را با i عوض می کند

1	2	4	6	5	3	7	8

۲- سپس یک خانه ی دیگر و همینطور ادامه دارد

1	2	4	5	6	3	7	8

۳- تا زمانی که به j برسد سپس j را

1	2	4	5	3	6	7	8

به اندازه ی $j-i$ جابجا می کنیم

1	2	4	3	5	6	7	8

1	2	3	4	5	6	7	8

Phase 1:

$j-i-1$ movements (switch)

Phase 2:

$j-i$ movements (switch)

$2j-2i-1$ switches → linear relation

اما این تعداد switch ها برای رسیدن به حالت sorted است اما خانه ی j ام باید قبل از خانه ی i ام قرار بگیرد با توجه به اینکه خانه های قبلی sorted هستند پس در کل $j-i$ نابجایی وجود دارد

نگین خیرمند ۹۸۳۱۰۲۳

برای سادگی بعضی از خط ها از دیگری جدا شدند اما تاثیری در order time ندارد،

۲-

`x=0`	1	C_1
`For(i=1; i≤n; i++){`	$n+1$	C_2
`for(j=1; j≤n; j++){`	$n(n+1)$	C_3
`x++`	$n\times n$	C_4
`}`		
`j=1`	n	C_5
`while(j<n){`	$(\log_2^n+1)\times n$	C_6
`x++;`	$(\log_2^n)\times n$	C_7
`j=j*2`	$(\log_2^n)\times n$	C_8
`}`		
`}`		

$$\sum_{\text{all statement}} (\text{cost of each statement})(\text{number of times executed})$$

$$= C_1\times 1 + C_2\times(n+1) + C_3\times(n)(n+1) + C_4(n\times n) + C_5\times n +$$

$$C_6\times(n)\times(\log_2^n+1) + C_7(\log_2^n\times n) + C_8\times(\log_2^n\times n)$$

$$= C_1 + nC_2 + C_2 + n^2C_3 + nC_3 + n^2C_4 + nC_5 + n\log_2^n C_6 +$$

$$nC_6 + n\log_2^n C_7 + n\log_2^n C_8 =$$

$$n^2\overbrace{(C_3+C_4)}^{a} + n\log_2^n\overbrace{(C_6+C_7+C_8)}^{b} + n\overbrace{(C_2+C_3+C_5+C_6)}^{c}$$

$$+\underbrace{(C_1+C_2)}_{d} = an^2 + bn\log_2^n + cn + d$$

$$T(n) = an^2 + bn\log_2^n + cn + d$$

T is a Function of n^2

۳_ بدترین حالت: عدد a در index=n (آخرین خانه‌ی آرایه) باشد

$\overbrace{i=1}^{c_0}\ ;\ \overbrace{i<=n}^{c_1}\ ;\ \overbrace{i++}^{c_2}$

	cost	تعداد تکرار
`For(i=1; i<=n; i++){`	c_i	n+1 ← شرط
`if(A[i]==a){`	c_3	n
`return i;`	c_4	1
`}`		
`}`		

$$\underbrace{C_0\times1}_{\text{مقداردهی } i} + \underbrace{C_1\times(n+1)}_{\text{چک کردن شرط For}} + \underbrace{C_2\times n}_{i++} + \underbrace{C_3\times n}_{\text{چک کردن if}} + \underbrace{C_4\times1}_{\text{return کردن}}$$

$$= (C_1+C_2+C_3)n + (C_0+C_1+C_4) = an+b = O(n)$$

بهترین حالت: عدد a در index=1 (اولین خانه‌ی آرایه باشد):

$\overbrace{i=1}^{c_0}\ ;\ \overbrace{i<=n}^{c_1}\ ;\ \overbrace{i++}^{c_2}$

	cost	تعداد تکرار
`For(i=1; i<=n; i++){`	c_i	1 ← شرط
`if(A[i]==a){`	c_3	1
`return i;`	c_4	1
`}`		
`}`		

$$C_0\times1 + C_1\times1 + C_2\times0 + C_3\times1 + C_4\times1$$

$$= (C_0+C_1+C_2+C_3+C_4) = b = O(1)$$

Average case Time = $\frac{\text{All possible case time}}{\text{num of cases}}$ حالت متوسط :

$$x + 3x + 6x = 10x = 1$$

$$\underbrace{\quad}_{x}\ \underbrace{\quad}_{3x}\ \underbrace{\quad}_{6x}$$

$\lfloor \frac{n}{3} \rfloor$ $\lfloor \frac{n}{3} \rfloor$ $\lfloor \frac{n}{3} \rfloor$

$$x = \frac{1}{10}$$

برای درایه ی x ام

	cost	number
For (i=1 ; i<=n ; i++){	$C_i \in \{0,1,2\}$	$n+1$: شرط
if(A[i] == a){	C_3	n
return i;	C_4	1
}		
}		

$$C_0 \times 1 + C_1 \times (n+1) + C_2 \times (n) + C_3 \times n + C_4$$

$$= (C_1 + C_2 + C_3)\,n + (C_0 + C_1 + C_4) = an + b$$

$$S_n = \frac{n}{2}(a_1 + a_n) \qquad : n \in \left[2 ; \lfloor \tfrac{n}{3} \rfloor\right]$$

$$S_n = \frac{1}{2} \times \left(\lfloor \tfrac{n}{3} \rfloor - 1\right)\left(2a + b + \lfloor \tfrac{n}{3} \rfloor a + b\right)$$

برای $i = 1$

A = Time Taken: $b + \frac{1}{2} \times \left(\lfloor \frac{n}{3} \rfloor - 1\right)\left((2 + \lfloor \frac{n}{3} \rfloor)a + 2b\right)$
for the first $\lfloor \frac{n}{3} \rfloor$

$$: n \in \left[\lfloor \tfrac{n}{3} \rfloor + 1 , 2\lfloor \tfrac{n}{3} \rfloor\right]$$

$$S_n = \frac{1}{2} \times \lfloor \tfrac{n}{3} \rfloor \times \left(a(\lfloor \tfrac{n}{3} \rfloor + 1) + b + 2a\lfloor \tfrac{n}{3} \rfloor + b\right)$$

Time Taken: $\frac{1}{2} \times \lfloor \frac{n}{3} \rfloor \times \left((3\lfloor \frac{n}{3} \rfloor + 1) \times a + 2b\right) = B$
For the second $\lfloor \frac{n}{3} \rfloor$

$$x \in \left[2\left\lfloor \frac{n}{3} \right\rfloor + 1 ; n\right]$$

$$S_n = \frac{1}{2} \times \left\lfloor \frac{n}{3} \right\rfloor \times \left(a\left(2\left\lfloor \frac{n}{3} \right\rfloor + 1\right) + b + an + b \right)$$

Time Taken for the third : $\frac{1}{2} \times \left\lfloor \frac{n}{3} \right\rfloor \times \left(\left(2\left\lfloor \frac{n}{3} \right\rfloor + 1 + n\right) a + 2b \right) = C$

$\left\lfloor \frac{n}{3} \right\rfloor$

Average Case Time: $a'A + b'B + c'C$

$a', b', c' \rightarrow$ are the probabilities

$\rightarrow \frac{1}{10} \times A + \frac{3}{10} \times B + \frac{6}{10} \times C$

4.a

```
... get n and k and elist                                   C1
For i in range(0, n):                                       (n+1) C2
    For j in range(i+1, n):                                 Σ_0^{n-1} (tj) C3
        if elist[i]+elist[j]==k                             Σ_0^{n-1} (tj-1) C4
            Print ("the list satisfies the condition")      C5 } فقط یک بار
            return                                          C6 } اجرا میشوند
Print ("the list does not satisfy the condition")           C7
return                                                      C8
```

$$\begin{Bmatrix} i=0 & j & 1\to n & n \\ i=1 & j & 2\to n & n-1 \\ \vdots & & & \vdots \\ i=n-1 & j & n & 1 \end{Bmatrix}$$

For درون: $\frac{n(n+1)}{2}$ For بیرون: $\frac{(n+1)(n+2)}{2}$

$$T(n) = C_1 + (n+1)C_2 + \sum_{0}^{n-1}(tj)C_3 + \sum_{0}^{n-1}(tj-1)C_4 + C_5 + C_6 + C_7 + C_8$$

$$T(n) = C_1 + (n+1)C_2 + \frac{n\times(n+1)}{2}C_4 + C_3\overset{n^2+3n+2}{\frac{(n+1)(n+2)}{2}} + C_5 + C_6 + C_7 + C_8$$

worst case scenario
↳ the list does not satisfy the condition

$$T(n) = C_1 + nC_2 + C_2 + \frac{n^2}{2}C_4 + \frac{n}{2}C_4 + \frac{n^2}{2}C_3 + \frac{3}{2}n\,C_3 + C_3 + C_5 + C_6 + C_7 + C_8$$

$$T(n) = n^2\left(\frac{C_4}{2} + \frac{C_3}{2}\right) + n\left(C_2 + \frac{C_4}{2} + \frac{3}{2}C_3\right) + \left(C_1 + C_2 + C_3 + C_5 + C_6 + C_7 + C_8\right)$$

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

$T(n) = c$
best case scenario → can find the a and b in the first and second index of the array

Code	Cost
... get n and elist andk	C_1
For i in range(0, n-1)	nC_2
b=k-elist[i]	$(n-1)C_3$
index= None	$(n-1)C_4$
start=0	$(n-1)C_5$
end= n-1	$(n-1)C_6$
mid = int((n-1)/2)	$(n-1)C_7$
while mid <=end :	$\sum_{j=0}^{n-1} t_j C_8$
if end - start<=1:	$\sum_{j=0}^{n-1} (t_j-1) C_9$
if elist[start] =b	C_{10}
index= start	C_{11}
elif elist [end] =b	C_{12}
index=b	C_{13}
break	C_{14}
mid = int ((end + start)/2)	$\sum_{j=0}^{n-1} (t_j-1) C_{15}$
if elist [mid]>b	$\sum_{j=0}^{n-1} (t_j-1) C_{16}$
end = mid	$\sum_{j=0}^{n-1} (t_j-1) C_{17}$
else:	$\sum_{j=0}^{n-1} (t_j-1) C_{18}$
start =mid	$\sum_{j=0}^{n-1} (t_j-1) C_{19}$
if index != None:	$\sum_{j=0}^{n-1} (t_j-1) C_{20}$
Print("this list has the condition")	C_{21}
Print("{}+{} = {}".format(elist[i] , b, k))	C_{22}
return	C_{23}
Print ("this list does not satisfy the condition")	C_{24}

$$T(n) = c_1 + nc_2 + (n-1)\overbrace{(c_3 + c_4 + c_5 + c_6 + c_7)}^{K_1} + \sum_{j=0}^{n-1} t_j c_8 + \sum_{j=0}^{n-1} (t_j - 1) c_9 + \overbrace{(K_{10} + c_{11} + c_{12} + c_{13} + c_{14})}^{K_2}$$

$$+ \sum_{j=0}^{n-1} (t_j - 1) \underbrace{(c_{15} + (c_{16} + c_{17}) + ((c_{18} + c_{19}) + c_{20}))}_{K_3} + \underbrace{(c_{21} + c_{22} + c_{23} + c_{24})}_{K_4}$$

one of these two

$$T(n) = c_1 + nc_2 + (n-1)(K_1) + \log_2^n c_8 + (\log_2^n - 1) c_9 + K_2 + (\log_2^n) K_3 + K_4$$

worst case scenario

& the list does not satisfy the condition

برنامه 2 تا 2 تا تقسیم می‌شد
لیست را مثل یک درخت یک هر node
درآن دو شاخه دارند. سمت راست هست [mid, end]
روشون میده و شاخه ی چپ [star, mid] رو
depth درخت نشان دهنده ی $\log_2^n$ است
و تعداد مراحلی که باید در درخت پایین رفت.

$$T(n) = an + b\log^n + c$$

$$\log_2^n < n$$

$$T(n) = b\log^n + c'$$

best case scenario

$a = 1$

5- a) $\log(n!) = \log^{(n)} + \log^{(n-1)} + \dots + \log^{(2)} + \cancel{\log^{(1)}}$

$\forall x \in [1, n]: \quad x \leq n$ می دانیم :

$\log^{x} \leq \log^{n} \longrightarrow$ به ازای هر عدد $\log^{n}$ را جایگذاری می کنیم :

$\log^{(n)} + \dots + \log^{(1)} \leq \underbrace{\log^{(n)} + \dots + \log^{(n)}}_{n}$

$\log^{(n)} + \log^{(n-1)} + \dots + \log^{(1)} \leq \underbrace{n \log^{(n)}}_{\log^{(n)^{n}}}$

$\Rightarrow \log^{(n!)} = O(\log^{(n)^{n}})$

حال فقط باید اثبات کنیم: $\log(n!) = \Omega(\log(n^n))$

یعنی $\exists c>0, n_0>0 / \forall n \geq n_0 : \log(n!) \geq c\log(n^n)$

فرض: $c = \frac{1}{2}$ $\quad \log(n!) \geq \frac{1}{2} \times \log(n^n)$

با روی استقرا $n_0 = 1$ $\quad \rightarrow n! \geq \sqrt{n^n}$

به توان ۲: $\quad n! \times n! \geq n^n \xrightarrow[\times (n+1)^2]{} ((n+1)!)^2 \geq n^n \times (n+1)^2$

می‌خواهیم ثابت کنیم: $\quad n^n (n+1)^2 \geq (n+1)^{n+1}$

تا ثابت شود:

$$(n+1)^{n+1} = (n+1)^2 \times (n+1)^{n-1} = \frac{(n+1)^2 \times (n+1)^n}{(n+1)} \rightarrow n^n \geq \frac{(n+1)^n}{n+1}$$

حال ثابت شد $n^n \geq \frac{(n+1)^n}{n+1}$

$$n \geq \frac{n+1}{\sqrt[n]{n+1}} \longleftrightarrow \frac{n+1}{n} \leq \sqrt[n]{n+1} \longleftrightarrow 1+\frac{1}{n} \leq \sqrt[n]{n+1}$$

$$\lim_{n \to \infty} \left(1+\frac{1}{n}\right)^n \leq \lim_{n \to \infty} \left(\sqrt[n]{n+1}^{\,n}\right) \quad \leftrightarrow$$

$$\leftrightarrow \quad e \leq \infty$$

$$((n+1)!)^2 \geq n^n (n+1)^2 \geq (n+1)^{n+1}$$

$\rightarrow (n+1)! \geq \sqrt{(n+1)^{n+1}}$ پایان استقرا

$$\log(n!) = \Omega(\lg(n^n))$$

$$\left.\begin{array}{l}\log(n!) = O(\log(n^n)) \\ \log(n!) = \Omega(\log(n^n))\end{array}\right\} \rightarrow \log(n!) = \Theta(\log(n^n))$$

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

$$n^{\log_n^2} = 2^{\log_n^n} = 2$$

I) $2 \overset{?}{=} O(1)$ کافیست c را بزرگتر از 2 انتخاب کنیم:

$\exists\, c>0, n_0>0 \;;\; \forall n \geq n_0 \quad 2 \leq c \times g(n) \rightarrow g(n)=1$

II) $2 = \Omega(1)$

$\exists c>0 \;;\; n_0>0 \;;\; \forall n \geq n_0 \quad 2 \geq c \times g(n) \rightarrow g(n) = 1$ ✓

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

$$\lim_{n\to\infty} \frac{2^n}{n!} \underset{\text{تقریب استرلینگ}}{=} \lim_{n\to\infty} \frac{2^n}{\sqrt{2\pi}\,(n^{\frac{1}{2}} + \frac{\alpha}{n^{\frac{1}{2}}})(\frac{n}{e})^n}$$

$$= \lim_{n\to\infty} \frac{(2e)^n}{\sqrt{2\pi}\; n^{\frac{1}{2}} \times n^n} = \lim_{n\to\infty} \frac{(\frac{2e}{n})^n}{\sqrt{2\pi n}} = 0$$

$$n! = \omega(2^n) \quad ✓$$

d) $n! = O(n^n)$

$n! \rightarrow$ log میگیریم $\rightarrow \log^{n!} = \log^1 + \log^2 + \dots + \log^n$

$n \to \infty$
$n > 1$
میدانیم $\rightarrow \begin{cases} \log^{(1)} \leq \log^{(n)} \\ \vdots \\ \log^{(n)} \leq \log^{(n)} \end{cases}$ $\quad : \lim_{n\to\infty} \frac{\log^{n!}}{\log^{n^n}} = 0$

$$\log^{n!} \leq \log^{n^n} \Longrightarrow n! \leq n^n \quad (n \to \infty)$$

$$n! = O(n^n)$$ ✓

با استفاده از استرلینگ:

تقریب استرلینگ

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

$$= \lim_{n\to\infty} \frac{1}{\sqrt{2\pi}\,\frac{n^{\frac{1}{2}}}{e^n}} = \lim_{n\to\infty} \frac{e^n}{\sqrt{2\pi n}} = \infty$$

رشد e^n بیشتر از $\sqrt{n}$ است.

a) $F(n) = O(g(n)) \Rightarrow f(n) \leq a \times g(n)$ -6

but if $a > 1 \Rightarrow g(n) \cancel{\leq} a' \times F(n)$ برقرار نیست

$F(n) = n^4$

$n^4 = O(n^5)$ ولی $n^5 \neq O(n^4)$

b) $F(n) = n^4$ max باید باشد ← برقرار نیست

$g(n) = n^5$

$F(n) + g(n) = O(n^5)$

c) $F(n) = O(g(n)) \xrightarrow{①} \lg(F(n)) = O(\lg(\lg(g(n))))$ نادرست

② $\forall n: \lg(g(n)) \geq 1$ and $F(n) \geq 1$

$F(n) = O(g(n)) \Rightarrow F(n) \leq g(n) \longrightarrow \lg(F(n)) \leq \lg(g(n))$

$\lg(F(n)) = O(\lg(g(n)))$ ① lg معکوس

اگر بتوانیم n=1 در نظر بگیریم، می‌توان g(n) و F(n) را به صورت زیر در نظر گرفت:

$F(n) = n, \; g(n) = n^2$ F $n = 1 \rightarrow \lg(\lg(1)) = 0 < 1$

② نقض شد.

S

d) $F(n) = O(g(n)) \longrightarrow 2^{F(n)} = O(2^{g(n)})$ درست

$F(n) \leq g(n) \xrightarrow[\text{صعودی } 2^0]{} 2^{F(n)} \leq 2^{g(n)} \longrightarrow 2^{F(n)} = O(2^{g(n)})$

e) $F(n) = \frac{1}{n} \rightarrow F(n^2) = \frac{1}{n^2}$

$F(n) \neq O\left(\frac{1}{n^2}\right) \longrightarrow$ برقرار نیست.

f) طبق ریاضی ۱: اگر تابع $g(n)$ حد بالای تابع $F(n)$ ← صحیح
باشد آنگاه $F(n)$ حد پایین $g(n)$ است (در دامنه مشترک)

$\exists\, c>0,\ n_0>0\ ;\ \forall n \geq n_0 \quad \boxed{F(n) \leq c\,g(n)} \xrightarrow{\times \frac{1}{c}} \frac{1}{c}F(n) \leq g(n)$

$\rightarrow g(n) \geq c' F(n)$

$\exists\, c'>0,\ n_0>0\ \forall n \geq n_0 :\ g(n) \geq c' F(n) \Rightarrow g(n) = \Omega(F(n))$

g) $F(n) = \Theta(F(\frac{n}{2}))$

$$\left.\begin{aligned} F(n) &= \frac{1}{n} \rightarrow O(\frac{1}{n}) \\ F(n) &= \frac{1}{n} \rightarrow \Omega(\frac{1}{n}) \end{aligned}\right\} \rightarrow \Theta(\frac{1}{n}) \neq \Theta(\frac{2}{n})$$

h) $F(n) + o(F(n)) = F(n) + c\,F(n) = (c+1)F(n)$

$F(n) \leq F(n) + o(F(n)) \overset{c>1}{\leq} c' F(n)$

$= \Theta(F(n))$

10) $\log^n \times \log(\log(n)) \ominus \log^n \times \log(\log(n))$

-7

	A	B	O	o	Ω	ω	Θ
1	n^2	n^3	yes	yes	no	no	no
2	$\log^k n$	n^ε	yes	yes	no	no	no
3	n^k	c^n	yes	yes	no	no	no
4	2^n	$2^{n/2}$	no	no	yes	yes	no
5	$n^{\lg c}$	$c^{\log^n}$	yes	no	yes	no	yes
6	$4^{\log^n}$	n^2	yes	no	yes	no	yes
7	$n!$	$n.2^n$	no	no	yes	yes	no
8	$\sqrt{2}^{\log^n}$	$2^{\sqrt{2\log^n}}$	no	no	yes	yes	no
9	$(\lg(n))!$	2^{2^n}	yes	yes	no	no	no
10	$n^{\lg(\lg(n))}$	$(\lg(n))^{\lg^n}$	yes	no	yes	no	yes

1) $n^2 = O(n^3)$ ✓ $n^2 = o(n^3)$ ✓ $n^2 = \Omega(n^3)$

2) $\log^k_n = \frac{\log^k}{\log^n} = \frac{c}{n} = cn^{-1} = O(n^3)$ ✓ $o(n^3)$ ✓

$= \Omega(n^\varepsilon)$ ∝

3) $n^k = O(c^n)$ ✓ $o(c^n)$ ✓ $\Omega(c^n)$ ∝ $\Theta(c^n)$ ∝

4) $2^n = O(2^{n/2})$ ∝

5) $n^{\lg c} \bigcirc c^{\log^n} \to \log^n \times \log^c \ominus \log^c \times \log^n$

6) $4^{\log^n} \bigcirc n^2 \to \log^4 \times \log^n \ominus 2\log^n$ $(n \to \infty)$

7) $\lim \frac{n \cdot 2^n}{n!} = \frac{2^n}{(n-1)!} = 0$

8) $2^{\frac{1}{2}\log^n} \bigcirc 2^{2\log^n}$

$\frac{1}{2}\log^n \bigcirc \sqrt{2\log^n}$

9) $(\log(n))! \bigcirc 2^{2^n}$

$\log^{(\log(n))!} \bigcirc \log^2 \times 2^n \to \log^{(\log(n))} + \log^{(\log(n)-1)} + \cdots \bigcirc k^n$