فرهاد امان ۹۹۳۱۰۰۶

۱- الف)

$\boxed{m = \text{طول آرایه } B}$

`int j = 0;`	c_1
`for(int i=0; i<n; i++){`	nc_2
`if (A[i]!= B[j]){`	c_3
`int k = j;`	c_4
`while(A[i]!= B[k]){`	mc_5
`k++;`	c_6
`}`	—
`swap(B[j], B[k]);`	c_7
`}`	—
`j++;`	c_8
`}`	—

$$nc_2 \times (c_3 + c_4 + mc_5 \times (c_6) + c_7 + c_8)$$

$$= nc_2 \times mc_5 = \boxed{n \times m}$$

از $O(n \times m)$ است.

ب) اگر n سایز آرایه A و m سایز آرایه B باشد. در واقع الگوریتم بالا تمامی اعضای A را در B پیدا می‌کند و آن‌ها را به ترتیب که در A وجود دارند در n عنصر اول آرایه B می‌چیند توجه کنید که باید $m \geq n$ باشد و همچنین تمام اعضای A حداقل یک بار در B ظاهر شده باشند وگرنه الگوریتم به مشکل خواهد خورد. اگر سایز هر دو آرایه یکسان باشد و دارای اعضای یکسان باشند پس از اجرای الگوریتم آرایه B به شکل آرایه A در خواهد آمد.

فرهاد امان ۹۹۳۱۰۶

۲ ـ ابتدا بر طبق یک تقریب می دانیم که

$$1+\frac{1}{2}+\frac{1}{3}+\dots+\frac{1}{n} \approx \log n = O(\log n)$$

الگوریتم تنها از یک حلقه تشکیل شده است که دقیقا n بار اجرا خواهد شد.

در هر بار اجرای حلقه $\frac{1}{i}$ احتمال وجود دارد که Hello World چاپ شود به جز $i=0$ که این احتمال ۰ است.

حالا اگر از اصل امید ریاضی استفاده کنیم. تعداد دفعات چاپ شدن به طور میانگین $1+\frac{1}{2}+\frac{1}{3}+\dots+\frac{1}{n}$ خواهد بود.

که همانطور که می دانیم از $O(\log n)$ است.

فرهاد امان ۹۹۳۱۰۰۶

۳ـ الف) برای این کار از یک الگوریتم جستجوی خطی ساده استفاده می‌کنیم
این الگوریتم از خانه ۰ آرایه شروع به حرکت کرده و به ترتیب تمام خانه‌های آرایه را
برای پیدا کردن عدد مورد نظر جستجو می‌کند.

```
for i = 0 to i = n-1 {          n
    if A[i] == T {              C1
        print: T -> i           C2
        end
    }
}
```

الگوریتم تنها دارای یک حلقه است که این
حلقه حداکثر n بار اجرا خواهد شد
و بقیه دستورات دارای زمان ثابت هستند
پس الگوریتم با $O(n)$ کار می‌کند.

ب) ج) دو فرض برای سؤال داریم اول اینکه نمی‌دانیم مرتب سازی براساس کدام
index انجام شده و دوم اینکه می‌دانیم در حالت اول ابتدا باید index را پیدا کنیم
ادامه الگوریتم مشابه خواهد بود.
پیدا کردن index مرتب سازی: می‌دانیم آرایه از دو بخش تشکیل شده است که هر بخش
به صورت صعودی مرتب شده و اعداد بخش اول بزرگتر از بخش دوم اند.
خانه اول آرایه را به عنوان کلید در نظر می‌گیریم حال بر روی آرایه یک binary search
ساده می‌زنیم به این شکل که هر بار آرایه را نصف می‌کنیم اگر عنصر وسط بزرگتر از
کلید بود به نیمه راست و اگر کوچکتر بود به نیمه چپ مراجعه می‌کنیم.
پس از پایان این binary search ما توانسته‌ایم index مرتب سازی را پیدا کنیم.
$O(\lg n)$
پیدا کردن جواب: حالا که index را پیدا کرده‌ایم آرایه را به دو بخش جدا تقسیم می‌کنیم.
ورودی هر بخش به صورت جدا یک binary search ساده می‌زنیم زیرا هر بخش
مرتب شده است هر binary search از $O(\lg n)$ خواهد بود.
در انتها از ۳ باینری سرچ استفاده کرده‌ایم که در کل $O(\lg n)$ می‌شود.

فرهاد امان ۹۹۳۱۰۰۶

۴- الف) اولاً می دانیم که کد دارای ۲ حلقه است هر کدام از حلقه ها صرف نظر از بهترین حالت بودن - بدترین حالت بودن - میانگین بودن به تعداد مشخص بار اجرا می شود نتیجه می گیریم تفاوتی میان این ۳ حالت در پیچیدگی زمانی نیست

```
for i <- n down to 1:            n
    for j <- 1 to i:             n
        if A[j] > A[j+1]:        c
            swap (A[j], A[j+1])  c
        end if
```

$$1+2+3+\dots+n = \frac{n(n+1)}{2} = \boxed{O(n^2)}$$

پس الگوریتم از $O(n^2)$ است.

ب) بهترین حالت زمانی اتفاق می افتد که آرایه مرتب شده باشد.

پس در هر بار اجرای حلقه درونی تعداد swap های انجام گرفته را می شماریم اگر این تعداد برابر ۰ بود متوجه می شویم که آرایه مرتب شده است و برنامه را خاتمه می دهیم.

```
for i <- n down to 1:
    cnt <- 0
    for j <- 1 to i:
        if A[j] > A[j+1]:
            swap (A[j], A[j+1])
            cnt <- cnt + 1
        end if
    if cnt == 0:
        end
    endif
```

فرهاد امان ۹۹۳۱۰۰۶

d - a) از تغییر متغیر $n = 2^x$ استفاده می کنیم.

$$(2^x)^{\frac{1}{\lg 2^x}} = (2^x)^{\frac{1}{x \lg 2}} = (2^x)^{\frac{1}{x}} = \boxed{2 = \Theta(1)}$$

b)

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

$$= \lim_{n\to\infty} \underbrace{\frac{1}{\sqrt{2\pi n}\left(1+\Theta\left(\frac{1}{n}\right)\right)}}_{\leq 1} \times \left(\frac{2e}{n}\right)^n \leq \lim_{n\to\infty} \underbrace{\left(\frac{2e}{n}\right)}_{< \frac{1}{2}}^n \leq \lim_{n\to\infty} \left(\frac{1}{2}\right)^n = \boxed{0}$$

c)

$$\lim_{n\to\infty} \frac{n^n}{n!} \overset{\text{تقریب استرلینگ}}{=} \lim_{n\to\infty} \frac{e^n}{\sqrt{2\pi n}\left(1+\Theta\left(\frac{1}{n}\right)\right)} \geq \lim_{n\to\infty} \frac{e^n}{c_1\sqrt{n}}$$

$$\geq \lim_{n\to\infty} \frac{e^n}{c_0 n} = \boxed{\infty}$$

فرهاد امان 9810069

۴- a) غلط $1 = O(n)$ و $n \neq O(1)$

b) غلط $2n = O(n)$ و $4^n \neq O(2^n)$

c) غلط $\frac{1}{n} \neq O(\frac{1}{n^2})$

d) درست از تعریف O و Ω استفاده می‌کنیم.

$$\exists c>0, n_0>0 \mid \forall n \geq n_0 : 0 \leq f(n) \leq c\,g(n) \Rightarrow \frac{1}{c} f(n) \leq g(n)$$

$$\Rightarrow \boxed{g(n) = \Omega(f(n))}$$

e) غلط $4^n \neq \Theta(2^n)$

f) درست از تعریف o و Θ استفاده می‌کنیم.

$$\exists c, n_0 > 0 \mid \forall n \geq n_0 \Rightarrow 0 \leq o(f(n)) \leq c \cdot f(n)$$

$$\exists c_1, c_2, n_0 > 0 \mid \forall n \geq n_0 \Rightarrow c_1 f(n) \leq f(n) + o(f(n)) \leq c_2 f(n)$$

فرهاد امان ۹۹۲۱۰۰۶

۷ -

y = yes N = no

O	o	Ω	ω	θ	A	B
y	y	N	N	N	n^2	n^3
y	y	N	N	N	$\lg^k n$	n^ε
N	N	y	y	N	n^k	c^n
N	N	y	y	N	2^n	$2^{n/2}$
y	N	y	N	y	$2^{2\lg n}$	n^2
N	N	y	y	N	$n!$	$n2^n$
y	N	y	N	y	$n^{\lg\lg n}$	$(\lg n)^{\lg n}$
N	N	y	y	N	$n^{\lg n}$	$(\lg n)^2$
N	N	y	y	N	$\frac{n^2}{\lg n}$	$n(\lg n)^2$
y	y	N	N	~~N~~	$n^{\frac{1}{n}}$	$\sqrt{2}^{\lg n}$