علی تقوی ۹۶۳۱۰۷۵ تمرین ۱ DS

۱)

$\log^{1000} n < n^{\frac{1}{n}} < 2^{\sqrt{\log n}} < n^{\frac{1}{\log n}} < \log n = \ln n < \log^{\sqrt{2}} n < \sqrt{2}^{\log n} < F_{\log n} <$ ادامه

ادامه $\rightarrow\ < \sqrt{n} < \log F_n = n < n\log n < n^{\sqrt{2}} < n^2 < F_{\frac{n}{2}} < F_n < 2^n$

۲)

Insertion Sort	تعداد تکرار × هزینه
for j=2 to n	$c_1 n$
key = A[j]	$c_2(n-1)$
i = j-1	$c_3(n-1)$
while i > 0 and A[i] > key	$c_4 \sum_{j=2}^{n} t_j$
A[i+1] = A[i]	$c_5 \sum_{j=2}^{n} (t_j - 1)$
i = i-1	$c_6 \sum_{j=2}^{n} (t_j - 1)$
A[i+1] = key	$c_7 (n-1)$

$$\Rightarrow T(n) = c_1 n + c_2(n-1) + c_3(n-1) + c_4 \sum_{j=2}^{n} t_j + c_5 \sum_{j=2}^{n} (t_j - 1) + c_6 \sum_{j=2}^{n} (t_j - 1) + c_7(n-1)$$

الف) بهترین حالت ← آرایه مرتب ⇐ $t_j = 1$ ⇐ $T_n = an + b$ ← بهترین حالت

ب) بدترین حالت ← ترتیب برعکس ⇐ $t_j = j$ ⇐

$$\begin{cases} \sum_{j=2}^{n} t_j = \sum_{j=2}^{n} j = \frac{(n-1)(n+2)}{2} \\ \sum_{j=2}^{n} (t_j - 1) = \sum_{j=2}^{n} (j-1) = \frac{n(n-1)}{2} \end{cases}$$

$\Rightarrow T(n) = an^2 + bn + c$ ← بدترین حالت

ج) حالت میانگین ← $t_j = \frac{j}{2}$ ⇐ $T(n) = an^2 + bn + c$ ← میانگین

۳

الف) ما باید نشان دهیم که عناصر 'A همان عناصر A هستند که جای آنها عوض شده است.

ب) ثابت حلقه برای خطوط ۱ تا ۴ برابر i است.

$length[A] = n$

این حلقه بازه‌ی اعداد صحیح ۱ تا n را پیمایش می‌کند. با توجه به اینکه n، طول آرایه، متناهی است

اگر ثابت کنیم که حلقه‌ی خطوط ۲ تا ۴ هم متناهی است، نتیجه این الگوریتم و حلقه‌هایش پایان‌پذیر هستند.

ثابت حلقه برای خطوط ۲ تا ۴ برابر j است که بازه‌ی اعداد صحیح n تا (i+1) را پیمایش می‌کند چون

$i \leq n$ و n نیز متناهی است، پس این حلقه‌ی خطوط ۲ تا ۴ هم پایان‌پذیر است، پس

نتیجه می‌گیریم که این الگوریتم و حلقه‌هایش پایان‌پذیر هستند.

ج) برای نشان دادن درستی این تساوی که در حقیقت نشان دادن درستی الگوریتم مرتب‌سازی است باید سه گام زیر را طی کنیم: (ثابت حلقه‌ی ما در اینجا برای خطوط ۱ تا ۴ همان i است)

شروع (Initialization): باید نشان دهیم ثابت حلقه قبل از اولین اجرای حلقه درست بوده است؛

در ابتدا $i = 1$ است و زیر آرایه‌ی $A[1..(i-1)]$ یک آرایه خالی است. چون نیاز به مرتب‌سازی ندارد،

پس ثابت حلقه برقرار است.

ادامه (Maintenance): این که اگر ثابت حلقه قبل از شروع یک تکرار حلقه درست باشد، تا قبل از شروع حلقه‌ی بعد نیز برقرار باقی می‌ماند.

وقتی که i به حلقه خطوط ۱ تا ۴ داده می‌شود، در ابتدا زیر آرایه $A[1..(i-1)]$ تشکیل شده است

از کوچکترین اعضای آرایه $A[1..n]$ البته در حالت مرتب‌شده. پس از پایان حلقه‌ی خطوط ۲ تا ۴.

$A[i]$ کوچکترین مقدار در زیرآرایه $A[i..n]$ است در نتیجه زیرآرایه $A[1..i]$ الان از کوچکترین

اعضای آرایه $A[1..n]$ در حالت مرتب‌شده، تشکیل شده است.

پایان (Termination): وقتی که $i = n$ شود، حلقه خطوط ۱ تا ۴ به پایان می‌رسد پس

زیرآرایه $A[1..(i-1)]$ که همان $A[1..(n-1)]$ است تشکیل شده از کوچکترین اعضای آرایه

$A[1..n]$ البته در حالت مرتب‌شده. یعنی فقط عضو $A[n]$ باقی می‌ماند که بزرگترین عضو است پس

آرایه $A[1..n]$ مرتب‌شده است پس داریم: $A[1] \leq A[2] \leq \dots \leq A[n]$

$$T(n) \leqslant c_1(n+1) + c_2\sum_{i=1}^{n}(n-i+1) + c_3\sum_{i=1}^{n}(n-i) = c_1(n+1) + c_2\frac{n}{2}(n+1) + c_3\frac{n}{2}(n-1)$$

بدترین حالت $\Rightarrow T(n)_{\text{حبابی}} = an^2 + bn + c$

الگوریتم مرتب سازی درجی در بهترین حالت بصورت $an+b$ است که از مرتب سازی حبابی که در هر صورت به شکل an^2+bn+c است، الگوریتم بهینه تری می باشد ولی در بدترین حالت هر دو الگوریتم به صورت an^2+bn+c هستند که از یک اوردر هستند.

④

الف) وقتی یک آرایه ۲ عضوی که مرتب هستند ورودی‌مان اند، الگوریتم odd-even sort از bubble sort بهتر عمل می‌کند پس:

length (آرایه) = ۲

A[1] = ۲

A[2] = ۳

ب) یک آرایه تک عضوی به طول آن در bubble sort بهینه تر از odd-even sort است، پس:

length (آرایه) = ۱

A[1] = ۵