۱-

الف) تابع دو آرایه A و B را دریافت می‌کند و یک آرایه جدید به نام C و به طول آرایه کوچکتر می‌سازد. سپس با یک حلقه for بر روی این آرایه‌ها حرکت می‌کند و در هر مرحله داریم: $C[i] = A[i]+B[i]$ و سپس آرایه C را باز می‌گرداند.
این آرایه از مرتبه $O(n)$ است و به حالات ورودی وابسته نیست.

ب) یک آرایه دریافت کرده و با یک حلقه for روی آن پیمایش می‌کند. ابتدا عنصر اول آرایه را یعنی $A[0]$ را در یک متغیر به نام Max می‌ریزد و در هر مرحله، عناصر بعدی آرایه را با Max مقایسه می‌کند. اگر از Max بزرگتر بوده آن عنصر را در Max می‌ریزد. این آرایه از $O(n)$ است و به حالت ورودی بستگی دارد. زیرا اگر آرایه‌ای که به تابع می‌دهیم مرتب شده باشد سریع عنصر Max را پیدا می‌کنیم.

ج) Binary Search این الگوریتم از $O(\log n)$ است و به حالت ورودی بستگی ندارد.

۲-

$$\log n \in O(\sqrt{n}): \quad \lim_{n\to\infty} \frac{\log n}{\sqrt{n}} \xrightarrow{Hop} \lim \frac{\frac{1}{n}}{\frac{1}{2\sqrt{n}}} = \lim_{n\to\infty} \frac{2\sqrt{n}}{n} = 0 \quad ✓$$

۳-

$$n^3 \in O(n!) \rightarrow \log n^3 \leq \log c + \underbrace{\log n!}_{n\log n} \rightarrow 3\log n \leq n\log n \quad ✓$$

۴- چک کردن زوج یا فرد بودن یک عدد - جمع و تفریق و ضرب و تقسیم متغیرها - دسترسی به یکی از عناصر آرایه

۵- بله اگر $A = O(B)$, $B = O(A)$ آنگاه داریم

$$c_1 B \leq A \leq c_2 B$$
$$c_1 A \leq B \leq c_2 A$$

یعنی $A = \Theta(B)$, $B = \Theta(A)$

همچنین داریم:

$$f(x) = A_0 + A_1 x + A_2 x^2 + \dots A_r x^r$$
$$g(x) = B_0 + B_1 x + B_2 x^2 + \dots B_r x^r$$

$$\rightarrow f(n) \in \Theta(g(n)) \rightarrow c_1 g(n) \leq f(n) \leq c_2 g(n)$$
$$\rightarrow C_1 = \frac{A_r}{B_r} - 1 \text{ و } C_2 = \frac{A_r}{B_r} + 1$$

$$c_1 f(n) \leq g(n) \leq c_2 f(n)$$

$$g(n) \in \Theta(f(n)) \rightarrow C_1 = \frac{B_r}{A_r} - 1 \text{ و } C_2 = \frac{B_r}{A_r} + 1$$

۶

$f(n) = f(n-1) + n^2 \rightarrow (x-1)(x-1)^3 = (x-1)^4$

$\rightarrow f(n) = c_1 + c_2 n + c_3 n^2 + c_4 n^3$

$f(1) = 1 = c_1 + c_2 + c_3 + c_4 = 1$

$f(2) = 5 \quad c_1 + 2c_2 + 4c_3 + 8c_4 = 5$

$f(3) = 14 \quad c_1 + 3c_2 + 9c_3 + 27c_4 = 14$

$f(4) = 30 \quad c_1 + 4c_2 + 16c_3 + 64c_4 = 30$

$\rightarrow c_2 + 3c_3 + 7c_4 = 4$

$\rightarrow c_2 + 5c_3 + 19c_4 = 9$

$\rightarrow c_2 + 7c_3 + 37c_4 = 16$

$\rightarrow 2c_3 + 12c_4 = 5$

$\rightarrow 2c_3 + 18c_4 = 7$

$\rightarrow 6c_4 = 2 \rightarrow c_4 = \frac{1}{3}, \; c_3 = \frac{1}{2}, \; c_2 = \frac{1}{6}, \; c_1 = 0$

$f(n) = \frac{1}{3}n^3 + \frac{1}{2}n^2 + \frac{1}{6}n = \frac{n}{6}(2n^2 + 3n + 1) = \frac{n(n+1)(2n+1)}{6}$

۷

$T(n) = 2T(n-2) + 1 \rightarrow (x-\sqrt{2})(x+\sqrt{2})(x-1)$

$O(n^{\frac{1}{2}}_{-1})$

$\rightarrow T(n) = c_1' \sqrt{2}^n + c_2'$

$\rightarrow T(0) = 0 \rightarrow c_1' + c_2' = 0$

$T(2) = 1 \rightarrow 2c_1' + 2c_2' = 1$

$\rightarrow c_1' = 1, \; c_2' = -1$

۸-

$T(n) = T(n-2) + 1 \rightarrow$ ~~[illegible]~~ $\begin{cases} \text{همگن} = (x-1) \\ \text{ناهمگن} : (x-1) \end{cases}$

$T(n) = c_1 + c_2 n \qquad T(0) = 0 \qquad T(0) = c_1 = 0 \rightarrow c_1 = 0$

$T(2) = 1 \qquad T(2) = c_1 + 2c_2 = 1 \qquad c_2 = \frac{1}{2}$

۹- مرتبه پیچیدگی در سوال ۷ برابر است با $n^{\frac{1}{2}}$ و در سوال ۸ برابر است با $\frac{1}{2}n$

$$T(n) = \frac{T(\sqrt{n})}{\sqrt{n}} + \frac{\log_2 n}{n} \rightarrow nT(n) = \sqrt{n}\,T(\sqrt{n}) + \log_2 n$$

$$f(n) = nT(n) \longrightarrow f(n) = f(\sqrt{n}) + \log_2 n \rightarrow n = 2^{2^k}$$

$$f(2^{2^k}) = f(2^{2^{k-1}}) + 2^k \rightarrow t(k) = t(k-1) + 2^k \rightarrow \begin{cases} \text{همگن} : (x-1) \\ \text{ناهمگن} : (x-2) \end{cases}$$

$$t(k) = c_1 + c_2 2^k \rightarrow f(n) = c_1 + c_2 \log_2 n$$

$$\rightarrow nT(n) = c_1 + c_2 \log_2 n \rightarrow T(n) = \frac{c_1}{n} + \frac{c_2 \log_2 n}{n}$$

$$T(2) = 1,\ T(4) = 1 \rightarrow \begin{matrix} T(2) = \frac{c_1}{2} + \frac{c_2}{2} = 1 \\ T(4) = \frac{c_1}{4} + \frac{2c_2}{4} = 1 \end{matrix} \rightarrow c_1 = 0,\ c_2 = 2$$

۱۲_ عناصر آرایه را دو به دو با هم مقایسه می‌کنیم. عنصر اول را با عنصر آخر مقایسه می‌کنیم و اگر عنصر اول کوچک‌تر بود، آن‌ها را جابه‌جا می‌کنیم سپس از سمت چپ یکی به جلو و از سمت راست یکی به عقب می‌رویم و این دو عنصر را با هم مقایسه می‌کنیم. این کار را تا زمانی که به وسط آرایه برسیم تکرار می‌کنیم. بعد از این کار، min تابع در نیمه اول آرایه قرار دارد و max آرایه در نیمه دوم آرایه قرار دارد. سپس در نیمه اول با $\frac{n}{2}-1$ مقایسه min را پیدا می‌کنیم و در نیمه دوم تابع با $\frac{n}{2}-1$ مقایسه max را پیدا می‌کنیم. در ابتدا نیز به $\frac{n}{2}$ مقایسه عناصر آرایه را با هم مقایسه کرده بودیم و در کل می‌شود $\frac{3n}{2}-2$ مقایسه.

۱۳_ عنصر اول آرایه A و B را در نظر می‌گیریم و با هم جمع می‌کنیم. اگر با مقدار مورد نظر برابر بود، مکان آن‌ها را باز می‌گردانیم. اگر از عدد مورد نظر کوچک‌تر بود، در آرایه A، یک خانه به جلو می‌رویم و سپس دوباره مقایسه می‌کنیم. اگر عدد مورد نظر از جمع آن‌ها بزرگ‌تر شد، در آرایه B یک خانه به جلو حرکت می‌کنیم. در بدترین حالت، باید هر دو آرایه را تا انتها پیمایش کنیم بنابراین پیچیدگی این الگوریتم برابر می‌شود با $O(m+n)$

۱۴ـ ارایه $i = 1$, $Count = 1$, Value

$A[0]$ را در Value قرار می‌دهیم. اگر $A[i]$ برابر Value باشد، Count را یکواحد اضافه می‌کنیم

و در غیر این صورت، یک واحد کم می‌کنیم

اگر Count برابر صفر شود، $A[i]$ را در Value قرار می‌دهیم و Count را یک واحد اضافه می‌کنیم.

این کار را تا زمانی که به انتهای آرایه برسیم ادامه می‌دهیم

سپس تعداد دفعات تکرار Value را با پیمایش بر روی آرایه می‌شماریم و اگر از $\frac{n}{2}$ بیشتر بود، Value

را به عنوان عنصر اصلی بازمی‌گردانیم. در غیر این صورت، آرایه عنصر اصلی ندارد.

۱۵ـ

$n\log n \in \Theta(\log n!) \rightarrow n\log n \leq c \log n!$

$\log n! = \log n + \log(n-1) + \log(n-2) + \dots + \log(3) + \log 2 + \log 1$

$\log n! = \log n + \log 2(n-1) + \log 3(n-2) + \dots + \log \frac{n}{2}(\frac{n}{2}+1) \rightarrow$ تعداد جملات $\frac{n}{2}$ است، هر جمله از $\log n$ بزرگتر است

$n\log n = \log n + \log n + \dots \rightarrow$ تعداد جملات n جمله است

$\log n + \log n + \dots \leq C_1(\log n + \log 2(n-1) + \log 3(n-2) + \log\frac{n}{2}(\frac{n}{2}+1)) \rightarrow C_1 = 2$

$n\log n \in O(\log n!)$

$\log n! \in O(n\log n) \rightarrow \log n! \leq C_2\, n\log n \rightarrow C_2 = \frac{1}{2} \rightarrow \log n! \in O(n\log n)$

$\rightarrow n\log n \in \Theta(\log n!)$

۱۶

$f(n) \in n^{O(1)} = f(n) = n^{g(n)} \rightarrow g(n) \in O(1)$

$$g(n) \leq c \rightarrow g'(n) \leq 0 \rightarrow \begin{cases} g'(n) = 0 \rightarrow g(n) = r\ ,\ r \in R \\ g'(n) < 0 \rightarrow g(n) = \frac{1}{n}, \frac{1}{n^2}, \frac{1}{\log n} \end{cases} \rightarrow f(n) = n^m \quad m \in N$$

ب) $f(n) \in o(n^{O(1)})$

(۱۷)

$$T(n) = \begin{cases} c_0 & n = n_0 \\ aT(\frac{n}{b}) + cn^k & n > n_0 \end{cases} \rightarrow T(n) = aT(\frac{n}{b}) + cn^k$$

$$n = b^l \rightarrow T(b^l) = aT(b^{l-1}) + cb^{lk} \rightarrow t(l) = at(l-1) + cb^{lk}$$

$$\begin{cases} \text{همگن} : (x-a) \\ \text{ناهمگن} : (x-b^k) \end{cases} \rightarrow \begin{matrix} t(l) = c_1 a^l + c_2 b^{lk} & a \neq b^k \\ t(l) = c_1 a^l + c_2 l a^l & a = b^k \end{matrix}$$

$$\rightarrow a > b^k \rightarrow t(l) = \underbrace{c_1 a^l}_{\text{جمله بزرگتر}} + c_2 b^{lk} \rightarrow c_1 a^l = c_1 a^{\log_b^n} = c_1 n^{\log_b^a}$$

$$T(n) \in O(n^{\log_b^a})$$

$$a < b^k \rightarrow T(n) = c_1 n^{\log_b^a} + \underbrace{c_2 n^k}_{\text{جمله بزرگتر}} \longrightarrow T(n) \in O(n^k)$$

$$a = b^k \rightarrow T(n) = c_1 n^{\log_b^a} + \underbrace{c_2 (\log_b^n)(a^{\log_b^n})}_{\text{جمله بزرگتر}} \rightarrow c_2 \log^n (b^k)^{\log_b^n} = c_2 n^k \log n$$

$$T(n) \in O(n^k \log n)$$

---

(۱۸) -

$$T(n) = \begin{cases} 2T(n-1) + 1 & n > 1 \\ 1 & n = 1 \end{cases} \rightarrow \begin{cases} \text{همگن} : (x-2) \\ \text{ناهمگن} : (x-1) \end{cases}$$

$$T(n) = c_1 2^n + c_2 \rightarrow \left. \begin{matrix} T(1) = 2c_1 + c_2 = 1 \\ T(2) = 4c_1 + c_2 = 3 \end{matrix} \right\} \rightarrow 2c_1 = 2 \rightarrow c_1 = 1 , c_2 = -1$$

$$2^n - 1 \in O(2^n)$$

$$T(n)\begin{cases}\dfrac{۸T^{۳}(\frac{n}{۲})}{T^{۲}(\frac{n}{۴})} & n>۲\\ ۱ & n=۱\\ ۲ & n=۲\end{cases} \rightarrow T(n)=\frac{۸T^{۳}(\frac{n}{۲})}{T^{۲}(\frac{n}{۴})} \rightarrow n=۲^{k} \rightarrow T(۲^{k})=\frac{۸T^{۳}(۲^{k-۱})}{T^{۲}(۲^{k-۲})}$$

$$\rightarrow t(k)=\frac{۸t^{۳}(k-۱)}{t^{۲}(k-۲)} \xrightarrow{\log} \log(t(k)) = ۳+۳\log(t(k-۱))-۲\log(t(k-۲))$$

$$f(k)=۳f(k-۱)-۲f(k-۲)+۳ \rightarrow \begin{cases}\text{همگن}: (x-۱)(x-۲)\\ \text{ناهمگن}: (x-۱)\end{cases}$$

$$\rightarrow f(k)=c_{۱}+c_{۲}k+c_{۳}۲^{k} \rightarrow \log(t(k))=c_{۱}+c_{۲}k+c_{۳}۲^{k} \rightarrow t(k)=۲^{c_{۱}+c_{۲}k+c_{۳}۲^{k}}$$

$$T(n)=۲^{c_{۱}+c_{۲}\log n+c_{۳}n}$$

$$\rightarrow T(۱)=۲^{c_{۱}+c_{۳}}=۱ \rightarrow c_{۱}+c_{۳}=۰$$

$$T(۲)=۲^{c_{۱}+c_{۲}+۲c_{۳}}=۲ \rightarrow c_{۱}+c_{۲}+۲c_{۳}=۱$$

$$T(۴)=۲^{c_{۱}+۲c_{۲}+۴c_{۳}}=۶۴ \rightarrow c_{۱}+۲c_{۲}+۴c_{۳}=۶$$

$$\rightarrow \begin{matrix}c_{۲}+c_{۳}=۱\\ c_{۲}+۲c_{۳}=۵\end{matrix} \rightarrow c_{۳}=۴ \,/\, c_{۲}=-۳ \,/\, c_{۱}=-۴ \rightarrow T(n)=۲^{-۴-۳\log n+۴n}$$

$$\rightarrow ۲^{۴(n-۱)}\times n^{-۳} \rightarrow \frac{۲^{۴(n-۱)}}{n^{۳}} \rightarrow O\left(\frac{۲^{۴(n-۱)}}{n^{۳}}\right)$$

۲۰ -

$n^{100}, n! \to \lim_{n\to\infty} \frac{n^{100}}{n!} = \lim_{n\to\infty} \frac{\overbrace{n\times n\times n\times\cdots\times n}^{\text{۱۰۰ تا}}}{(n)(1)\times(n-1)(2)\times(n-2)(3)\times\cdots\times(n-99)(100)\times(n-100)\times\cdots\times 101}$

$\to \lim_{n\to\infty} \frac{n}{n}\times\frac{n}{2(n-1)}\times\frac{n}{3(n-2)}\times\cdots\times\frac{n}{100(n-99)}\times\frac{1}{n-100}\times\cdots\times\frac{1}{101} = 0 \to n^{100}\in O(n!)$

a^n, n^{100} $a>1 \to$ بجای n^{100} ، a^n لگاریتم آنها را بررسی می‌کنیم $\to \lim_{n\to\infty} \frac{\log(n^{100})}{\log(a^n)} \to \lim_{n\to\infty} \frac{100}{\log a}\times\frac{\log n}{n} = 0$

$\to n^{100}\in O(a^n) \cdot a>1$

$(1.005)^n, n! \to (1.005)^n \le c\,n! \xrightarrow{\log} n\log(1.005) \le c'(\log n!)$

$\to \underbrace{\log(1.005)+\log(1.005)+\cdots+\log(1.005)}_{n \text{ تا}} \le c'(\underbrace{\log n+\log(n-1)+\log(n-2)+\cdots+\log 3+\log 2+\log 1}_{n})$

$\to n\log(1.005) \le c'(\underbrace{\log n+\log 2(n-1)+\log 3(n-2)+\cdots+\log\frac{n}{2}(\frac{n}{2}+1)}_{\frac{n}{2}})$

$c'=2 \to 1.005^n \in O(n!)$

$\boxed{(1.005)^n < n^{100} < n!}$