Subject:
Date

① الف) از عنصر اول تا آخر را می‌ریم، دونه دونه عنصرهای متناظر رو مقایسه می‌کنیم با هم ← وابسته به n

ب) از عنصر اول تا آخر مقایسه می‌کنیم (با max) هرکدوم بزرگتر بود max رو آپدیت می‌کنیم ← وابسته به n

ج) از عنصر اول تا آخر، با عدد خاص مقایسه می‌کنیم تا پیدا بشه ← وابسته به n

② $\log n \leq c\sqrt{n} \Rightarrow \frac{\log n}{\sqrt{n}} \leq c \Rightarrow \lim_{n\to\infty} \frac{\log n}{\sqrt{n}} \xrightarrow{\text{هوپیتال}} \lim \frac{1/n}{1/2\sqrt{n}} = \frac{2\sqrt{n}}{n}$

$\Rightarrow \lim_{n\to\infty} \frac{2}{n} = 0$

③ $n^3 \leq c(n!) \Rightarrow \frac{n^3}{n!} \leq c \Rightarrow n! = n(n-1)(n-2) — (3)(2)(1)$

④ $f(n) \leq c \times 1 \Rightarrow f'(n) \leq 0 \Rightarrow f(n) = 0$ (تابع ثابت)

$\Rightarrow f'(n) < 0 \longrightarrow f(n) = 1/n, 1/n^2 —$

⑤ $f(n) \in \Theta(g(n)) \Rightarrow c_1 g(n) \leq f(n) \leq c_2 g(n) \Rightarrow \begin{cases} c_1 = \frac{f(n)}{g(n)} + 1 \\ c_2 = \frac{f(n)}{g(n)} - 1 \end{cases}$

$g(n) \in \Theta(f(n)) \Rightarrow c_1 f(n) \leq g(n) \leq c_2 f(n) \Rightarrow \begin{cases} c_1 = \frac{g(n)}{f(n)} + 1 \\ c_2 = \frac{f(n)}{f(n)} - 1 \end{cases}$

۶) $f(n) = f(n-1)$

۷) وقتی $n=0$ تابع ۲ را بازمی‌گرداند ($O(1)$) و وقتی $n \neq 0$ تابع دوبار $f(n-2)$ را فراخوانی می‌کنه پس

$$T(n) = 2T(n-2) + O(1) \rightarrow T(n-2) = 2T(n-4) + O(1)$$

$$\rightarrow T(n-4) = 2T(n-6) + O(1) \Rightarrow k = \frac{n}{2} \Rightarrow T(n) = O(2^{n/2})$$

۸) در حالت مثل قبل یک $O(1)$ داریم ولی خود تابع یکبار فراخوانی می‌شود چون فراخوانی تنها یکیه از مرتبه $O(n)$ ـه

$$T(n) = T(n-2) + O(1)$$

۹) مرتبه پیچیدگی ۷ ← $O(2^n)$ و برای ۸ ← $O(n)$، چون توی ۷ دوبار فراخوانی می‌شه تعداد فراخوانی‌ها خیلی زیاد می‌شن ولی توی الگوریتم ۸ در یک فراخوانی ۲ مسئله رو ساده‌تر حل می‌کنه

۱۰)

$$T(n) = 2T(n-4) + T(n-2) + O(1)$$

$$T(n-2) = 2T(n-6) + T(n-4) + O(1) \xrightarrow{\text{جایگذاری}} T(n) = 2T(n-4) + 2T(n-6) + T(n-4) + O(1) + O(1)$$

$$\Rightarrow T(n) = 3T(n-4) + 2T(n-6) + O(1)$$

$$k = \frac{n}{4} \rightarrow 3^k = 3^{n/4} \rightarrow O(3^{n/4})$$

۱۱)

$$T(n) = \begin{cases} \frac{T(\sqrt{n})}{\sqrt{n}} + \frac{\log_2 n}{n} & n > 2 \\ 1 & n = 2 \end{cases} \rightarrow T(\sqrt{n}) = \frac{T(\sqrt[4]{n})}{\sqrt[4]{n}} + \frac{\log \sqrt{n}}{\sqrt{n}} \rightarrow \frac{1}{2}\log n$$

$$\Rightarrow T(n) = \frac{1}{\sqrt{n}}\left(\frac{T(\sqrt[4]{n})}{\sqrt[4]{n}} + \frac{\log n}{2\sqrt{n}}\right) + \frac{\log n}{n}$$

⑫ ① مقایسه همه عنصرها و جابجا کردن (همه اعداد کوچک‌تر تو نیمه اول و همه اعداد بزرگ‌تر تو نیمه دوم) $\frac{n}{2}$

② مقایسه نیمه اول و دوم برای پیدا کردن min و max (۲ - ۲ $\frac{n}{2}$)

$\frac{3n}{2}$ مقایسه

⑬ ① اولین عنصر A و B رو باهم جمع می‌کنیم $A[i] + B[j]$

② اگر با عدد داده برابر بود برمی‌گردونیم

③ اگر کمتر بود $A[i]$ رو باید بریم سمت راست تا بیشتر شه

④ اگر بیشتر بود $B[j]$ رو باید بریم سمت راست تا کمتر شه

⑤ وقتی یکی از i یا j به ته رسید تموم می‌شه

* می‌شه $O(m+n)$ چون یک بار A رو پیمایش می‌کنه یا یک بار B رو

⑭ ① اولین عنصر رو به عنوان عنصر فعلی می‌گیریم و به counter اضافه می‌کنیم

② اگر عنصر بعدی با عنصر اول برابر بود به counter اضافه می‌کنیم، اگر نبود کم می‌کنیم

③ اگر به یک عنصر رسیدیم و counter صفر بود، یکی بهش اضافه می‌کنیم

* $O(n)$ می‌شود و چون $\Omega(n)$ هست می‌توان گفت $\theta(n)$

⑮ $n \log n \leq c \log n! \Rightarrow \log n! = \log(n)(n-1)(n-2) \ldots (3)(2)(1) \Rightarrow \log \underbrace{(n)(2n-2) \ldots (\frac{n^2}{4})}_{\frac{n}{2}\ \text{جمله}}$

$\Rightarrow \log n \leq \log n,\ \log n \leq \log(2n-2) \Rightarrow c = 2$

⑯ الف) $n^{O(1)} \Rightarrow f(n) \in n^{O(1)} \Rightarrow f(n) = n^{g(n)} \Rightarrow g(n) \in O(1)$

$g(n) \leq c \Rightarrow g'(n) \leq 0 \Rightarrow f(n) = n^k$?

ب) $O(n^{O(1)}) = O(n^k)$,

ج)

Subject:
Date

$n=b^l \rightarrow T(b^l) = aT(b^{l-1}) + cb^{lk} \rightarrow t(l) = at(l-1) + cb^{lk} \rightarrow (b^k)^l$ (17)

$(x-a)(x-b^k)^{0+1} \Rightarrow x_{1,2} = a, b^k$

$t(l) = c_1 a^l + c_2 b^{kl}$ $b^k \neq a$

$t(l) = c_1 a^l + c_2 l a^l$ $b^k = a$ $\longrightarrow T(n) = O(n^k \log n)$

$n = b^l \longrightarrow l = \log_b n \longrightarrow T(n) = c_1 a^{\log_b^n} + c_2 (\log_b^n) a^{\log_b^n}$ $b^{\log_b n^k} = n^k$

$a > b^k$ $T(n) = O(n^{\log_b^a})$ $a < b^k$ $T(n) = O(b^{kl}) = O(b^{k\log_b^n}) = O(n^k)$

$T(n) = \begin{cases} 2T(n-1) + 1 \ (O(1)) & n > 1 \\ 1 & n = 1 \end{cases}$ $\begin{cases} \text{ریشه} & 1, (n-2) \\ \text{ریشه} & 1, (n-1) \end{cases}$ $\rightarrow c_1 2^n + c_2 \rightarrow 2c_1 + c_2 = 1$ (18)

$\rightarrow T(1) = 1, T(2) = 3 \rightarrow 4c_1 + c_2 = 3 \Rightarrow c_1 = 1, c_2 = -1$

ج) $2^n - 1 \in O(2^n)$

$T(n) = \begin{cases} \dfrac{8T^2(n/2)}{T^4(n/4)} & n > 2 \\ 1 & n = 1 \\ 2 & n = 2 \end{cases}$ $\Rightarrow n = 2^k \rightarrow \dfrac{8T^2(2^{k-1})}{T^4(2^{k-2})} = t(k)$ (19)

$\Rightarrow \log(t(k)) = \log\left(\dfrac{8t^2(2^{k-1})}{t^4(2^{k-2})}\right)$

$\lim_{n\to\infty} \dfrac{n^{100}}{n!} = \lim_{n\to\infty} \dfrac{n \times n \quad \times n}{n(1)(n-1)(2) \quad --- \quad (n-100) \quad -} = \dfrac{n}{n} \times \dfrac{n}{2(n-1)} \times \quad \times \dfrac{1}{(n-100)} \times \dfrac{1}{100}$ (20)

$\Rightarrow 0 \rightarrow$

$(4n-a)^n, n! \rightarrow \lim \dfrac{a^n}{n!}$, $a^n, n^{100} \rightarrow \lim \dfrac{\log(n^{100})}{\lg(a^n)} = \dfrac{100}{a} \times \dfrac{\log n}{n}$ $n^{100} \in O(n!)$ $\to 0$, $n^{100} \in O(a^n)$