Problem Set 1

جواب سؤال ۱: اول $T(n)$ را به دست می آوریم:

در هر مرحله که i تا از خانه‌های آرایه مرتب شده است حداکثر برای پیدا کردن جای خانه جدید ما به $\lg i$ تا مقایسه نیاز داریم. که در کل حداکثر تعداد مقایسه‌ها در این الگوریتم می‌شود:

$$\lg 1 + \lg 2 + \dots + \lg n = \lg 1\times 2\times 3\times \dots \times n = \lg n!$$

که در حالت بهترین یعنی مانند خود مرتب سازی درجی به n مقایسه نیاز داریم. ولی در حالت متوسط به ضریبی از $\lg n!$ می‌رسیم که آن ضریب کمتر از ۱ است و می‌توان گفت به طور متوسط ما به $c\lg n!$ هزینه نیاز داریم.

حالا می‌خواهیم اثبات کنیم که Θ، $\lg n!$ از $n\lg n$ است.

$$1 \le i \le n \Rightarrow \lg i < \lg n \Rightarrow \sum_{i=1}^{n} \lg i < n\lg n \Rightarrow \boxed{\lg n! < n\lg n}$$

$$\lg n! = O(n\lg n)$$

حالا برای آنکه اثبات کنیم که این الگوریتم از $\Theta(n\lg n)$ است باید اثبات کنیم که آن از $\Omega(n\lg n)$ نیز هست که آن را با تقریب استرلینگ استفاده می‌کنیم

$$\Rightarrow \Theta(n\lg n)$$

جواب سؤال ۲: چون ۲ را می‌توان ضریبی از 2^n دانست درست است ولی 2^{2n} چون ۲ به n وابسته است ضریب آن و ضریب هر توان ۲ می‌تواند بگیرد پس می‌توان گفت که جمله دوم غلط است. ($2^{2n} \neq O(2^n)$)

جواب سؤال ۳: اول قسمتی را اثبات می‌کنیم که اگر $O(g(n))$ و $\Omega(g(n))$ باشد آنگاه آن الگوریتم از $\Theta(g(n))$ است:

$$O(g(n)) \Rightarrow \exists c_1, n_0 ;\ n>n_0 \Rightarrow c_1 g(n) > T(n)$$

$$\Omega(g(n)) \Rightarrow \exists c_2, n'_0 ;\ n>n'_0 \Rightarrow c_2 g(n) < T(n)$$

$$\Rightarrow \exists c_1, c_2 ;\ n>k \Rightarrow c_1 g(n) > T(n) > c_2 g(n), \quad k=\max(n_0, n'_0)$$

$\Rightarrow$ الگوریتم ما از $\Theta(g(n))$ است.

حالا اثبات می‌کنیم که اگر الگوریتم از $\Theta(g(n))$ باشد آنگاه آن الگوریتم از $O(g(n))$ و $\Omega(g(n))$ است.

$$\Theta(g(n)) \Rightarrow \exists c_1, c_2, n_0 ;\ n>n_0 \Rightarrow c_1 g(n) > T(n) > c_2 g(n)$$

$\Rightarrow \exists c_2, n_0 ;\ T_n > c_2 g(n) \Rightarrow$ الگوریتم از $\Omega(g(n))$ است.

$\exists c_1, n_0 ;\ T_n < c_1 g(n) \Rightarrow$ الگوریتم از $O(g(n))$ است.

پس هر ۲ طرف اگر و فقط اگر اثبات شد.

جواب سئوال ۴: با استقراء اثبات می کنم.

$\lg n = m \qquad [\lg n]! = m!$

$\Rightarrow 2^m = n \Rightarrow$

$m! > n^k$

$m! > 2^{mk}$ فرض

$\lg(m+1)! > 2^{(m+1)k}$ حکم

اثبات حکم

$\Rightarrow (m+1)\,m! > 2^{mk} \times 2^k$

حتما از m ای به بعد درست در می آید. که آن m وابسته به k است چون 2^k ثابت است. $\Leftarrow m+1 > 2^k$

فرض $\Leftarrow m! > 2^{mk}$

$\Rightarrow (m+1)! > 2^{(m+1)k} \Rightarrow [\lg 2n]! > (2n)^k$

اثبات قسمت دوم:

$\lg\lg n = m$

$\Rightarrow 2^{2^m} = n \Rightarrow [\lg\lg n]! = m!$

$m! < n^k$

$m! < \left(2^{2^m}\right)^k$

$m! < 2^{k2^m}$ فرض:

$(m+1)! < 2^{k2^{m+1}}$ حکم:

$(m+1)\,m! < 2^{k2^m \times 2}$

فرض $\Leftarrow m! < 2^{k2^m}$

از جایی به بعد درست است $\Leftarrow m+1 < 2^{k2^m}$

$\Rightarrow (m+1)! < 2^{k2^{m+1}} \Rightarrow [\lg\lg 2n]! < (2n)^k$

جواب سئوال ۵: اگر ورودی همان را k بگیریم و آن شرط زیر را داشته باشد

$\underbrace{2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}}_{p} \le k < \underbrace{2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}}_{p+1}$

آنگاه $\lg^*$ همه این اعداد مثل k برابر با p خواهد بود که این یعنی از همه آنها یک بار lg بگیریم نتیجه برابر خواهد شد با lg p. و از طرف دیگر داریم

$\underbrace{2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}}_{p-1} \le \lg k < \underbrace{2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}}_{p}$

که اگر از آن $\lg^*$ بگیریم آنوقت حاصل آن عددی بین $p-1$ تا p خواهد شد.

$\Rightarrow p-1 \le \lg^*\lg n < p$

$\lg\lg^* n = \lg p$

$\lg p < p-1$ از جایی به بعد

$\Rightarrow \lg^*\lg n > \lg\lg^* n$

از جایی به بعد

جواب سوال ۶: a: اشتباه است:

$$\left.\begin{array}{l} f(n)=n^2 \\ g(n)=n \end{array}\right\} \Rightarrow n^2+n \neq \Theta(n)$$

مثال نقض

b) اشتباه است:

$$f(n)=1/n \Rightarrow 1/n \neq O(1/n^2)$$

مثال نقض

c) اشتباه است:

$$f(n)=2^n \Rightarrow 2^n \neq \Theta(\sqrt{2^n})$$

مثال نقض

d) درست است اثبات:

$$g(n)=o(f(n)) \Rightarrow f(n)=\omega(g(n))$$

$$\Rightarrow \exists c, n_0>0 ;\quad c\,g(n) < f(n)$$

$$\left.\begin{array}{l} \Rightarrow \exists c_2 ; \quad c_2(f(n)+g(n)) < f(n) \\ c=1, n>0 \Rightarrow f(n)+g(n) > f(n) \end{array}\right\} \Rightarrow \Theta(f(n)) = f(n)+o(f(n))$$

جواب سوال ۷:

$$T(n)=O(cn^{\log_3 4})$$

$$T(n)=4T(n/3)+n$$

$$\leq 4c(n/3)^{\log_3 4}+n$$

$$\leq 4c\,4^{\log_3 n/3}+n$$

$$\leq 4c\,4^{\log_3 n-\log_3 3}+n$$

$$\leq \frac{4c}{4}\, n^{\log_3 4}+n$$

$$\leq cn^{\log_3 4}+n \Rightarrow$$

این متناقض است

فرض $T(n)=\Omega(cn^{\log_3 4}-3d_1 n)$

$$T(n)=4T(n/3)+n \qquad d_1<1$$

$$\geq 4c(n/3)^{\log_3 4}-4d_1 n+n$$

$$\geq 4c(n/3)^{\log_3 4}-3d_1 n$$

$$\geq 4c\,4^{\log_3 n-1}-3d_1 n$$

$$\geq cn^{\log_3 4}-3d_1 n \Rightarrow \Omega(cn^{\log_3 4}-3d_1 n)$$

$$T(n)=O(cn^{\log_3 4}-3dn) \qquad d>1$$

$$T(n)=4T(n/3)+n$$

$$\leq 4c(n/3)^{\log_3 4}-4dn+n$$

$$\leq 4c(n/3)^{\log_3 4}-3dn$$

$$\leq 4c\,4^{\log_3 n/3}-3dn$$

$$\leq 4c\,4^{\log_3 n-1}-3dn$$

$$\leq \frac{4}{4}c\,n^{\log_3 4}-3dn$$

$$\leq cn^{\log_3 4}-3dn = O(cn^{\log_3 4}-3dn)$$

چون الگوریتم ما از $\Omega(cn^{\log_3 4}-3dn)$ و $O(cn^{\log_3 4}-3dn)$ است پس می‌توان گفت $T(n)$ از $\Theta(cn^{\log_3 4})$ است.

جواب سؤال ۸: فرض اول ما که باید اثبات کنیم از $\Theta(n^2)$ هست اشتباه است بیایید با قضیه اساس به حل این مسئله بپردازیم:

$$\log_b a = \log_2 4 = 2 \Rightarrow n^2 = \Theta(n^2)$$

چون جمله ما و $f(n)$ ما هر دو هم مرتبه اند پس باید جواب را در $\log n$ ضرب کرد که جواب ما از $T(n^2\log n)$ خواهد شد. حالا مسئله را برای $n^2\log n$ حساب می‌کنیم.

$$T(p) = O(c\,p^2\log p)$$
$$\begin{aligned}
T(n) &= 4T(n/2) + n^2 \\
&\le 4c(n/2)^2\log n/2 + n^2 \\
&\le 4c\,n^2/4\,(\log n - \log 2) + n^2 \\
&\le cn^2(\log n - 1) + n^2 \\
&\le cn^2\log n - cn^2 + n^2 \qquad c \ge 1 \\
&\le cn^2\log n
\end{aligned}$$

$$T(p) = \Omega(c\,p^2\log p)$$
$$\begin{aligned}
T(n) &= 4T(n/2) + n^2 \\
&\ge 4c_1(n/2)^2\log n/2 + n^2 \\
&\ge 4c_1 n^2/4\,(\log n - \log 2) + n^2 \\
&\ge c_1 n^2(\log n - 1) + n^2 \\
&\ge c_1 n^2\log n - c_1 n^2 + n^2 \\
&\ge c_1 n^2\log n \qquad c_1 \le 1
\end{aligned}$$

چون $T(n)$ هم از $O(n^2\log n)$ است و هم از $\Omega(n^2\log n)$ پس آن از $\Theta(n^2\log n)$ هم است.