复习 PPT 中第三章和第四章的例题答案：

1．Let $f(n)$ and $g(n)$be asymptotically nonnegative functions. Prove that $\Theta(f(n))+\Theta(g(n)) = \Theta(f(n)+g(n))$

Proof:

Assume that $\Theta(f(n)) = h(n)$ and $\Theta(g(n)) = p(n)$.

Then, $\Theta(f(n)) = h(n)$

$$\Longrightarrow \quad \exists c_1, c_2, n_0 > 0\ \forall n > n_0: c_2 f(n) \le h(n) \le c_1 f(n)$$

And $\Theta(g(n)) = p(n)$

$$\Longrightarrow \quad \exists c_3, c_4, n_1 > 0\ \forall n > n_1: c_4 g(n) \le p(n) \le c_3 g(n)$$

So, $\forall n > \max\{n_0, n_1\}$:

$$c_2 f(n) + c_4 g(n) \le h(n) + p(n) \le c_1 f(n) + c_3 g(n)$$

$$\Longrightarrow \min(c_2, c_4)(f(n) + g(n)) \le h(n) + p(n) \le \max(c_1 + c_3)(f(n) + g(n))$$

$$\Longrightarrow h(n) + p(n) = \Theta(f(n) + g(n))$$

$$\Longrightarrow \Theta(f(n)) + \Theta(g(n)) = \Theta(f(n) + g(n))$$

2. 用展开法求 T(n) = T(n-1) + n (n>1), T(1) = 1 的渐进时间复杂度 $f(n)$

T(n)=T(n-1)+n

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

=T(n-3)+n-1+n

=...

=T(1)+(2+3...+n-1+n) (according to T(1)=1)

=(1+2+3...+n-1+n)

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

$= \Theta(n^2)$

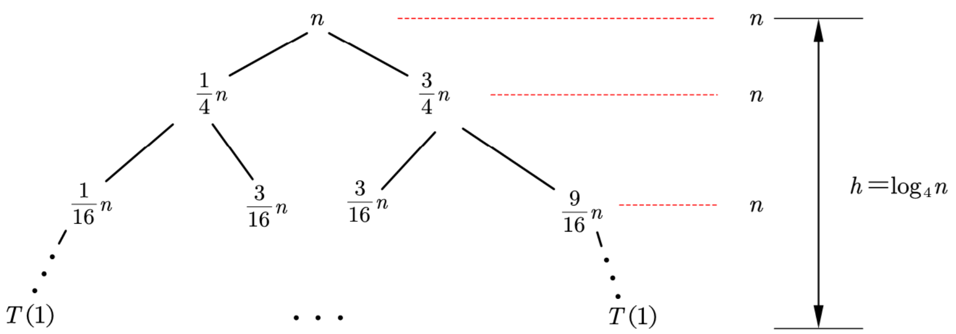
3. 合并排序过程中，对较长的数组按 2：1 比例划分，给出描述该排序算法时间复杂度的递推方程.

T(n) = T(2n/3)+T(n/3)+ θ(n)(或 n)

4. Draw the recursion tree (递归树) of the recurrent function:

T(n)=T(3n/4)+T(n/4)+ Θ(n), prove the tight bound (Θ) of your recurrence with substitution(替代法)

以下递归树，可得 T(n)= Θ(nlogn)



假设当$n \geqslant n_0$时 $T(n) \leqslant d_1 n \log n$

$$T(n) = T\left(\frac{1}{4}n\right) + T\left(\frac{3}{4}n\right) + \Theta(n)$$

$$\leqslant d_1 \frac{1}{4} n \log\left(\frac{1}{4}n\right) + d_1 \frac{3}{4} n \log\left(\frac{3}{4}n\right) + c\ n$$

$$= d_1 \frac{1}{4} n \log n + d_1 \frac{3}{4} n \log n + d_1 \frac{1}{4} n \log \frac{1}{4} + d_1 \frac{3}{4} n \log \frac{3}{4} + c\ n$$

$$= d_1 n \log n + \left(d_1\left(\frac{1}{4}\log\frac{1}{4} + \frac{3}{4}\log\frac{3}{4}\right) + c\right)n$$

当$n \geqslant n_0$时，只要$d_1 \geqslant \frac{-c}{\left(\frac{1}{4}\log\frac{1}{4} + \frac{3}{4}\log\frac{3}{4}\right)}$ $\quad T(n) \leqslant d_1 n \log n$ 所以$T(n) = O(n \log n)$

假设当$n \geqslant n_0$时 $T(n) \geqslant d_2 n \log n$

$$T(n) = T\left(\frac{1}{4}n\right) + T\left(\frac{3}{4}n\right) + \Theta(n)$$

$$\geqslant d_2 \frac{1}{4} n \log\left(\frac{1}{4}n\right) + d_2 \frac{3}{4} n \log\left(\frac{3}{4}n\right) + c\ n$$

$$= d_2 \frac{1}{4} n \log n + d_2 \frac{3}{4} n \log n + d_2 \frac{1}{4} n \log \frac{1}{4} + d_2 \frac{3}{4} n \log \frac{3}{4} + c\ n$$

$$= d_2 n \log n + \left(d_2\left(\frac{1}{4}\log\frac{1}{4} + \frac{3}{4}\log\frac{3}{4}\right) + c\right)n$$

当$n \geqslant n_0$时，只要$d_2 \leqslant \frac{-c}{\left(\frac{1}{4}\log\frac{1}{4} + \frac{3}{4}\log\frac{3}{4}\right)}$ $\quad T(n) \geqslant d_2 n \log n$ 所以$T(n) = \Omega(n \log n)$

所以$T(n) = \Theta(n \log n)$

PTA

动态规划练习题 1

1-1 如果一个问题可以用动态规划算法解决，则总是可以在多项式时间内解决的。答案：F

提示：采用动态规划求解的时间与 (# of subproblems overall) × (# of choices)相关，因此即使独立子问题的个数(# of subproblems overall)是多项式，但解决子问题需选择的数(# of choices)可能非多项式。

习题课（11 周）

1-3 哈夫曼编码是一种最优的前缀码。对一个给定的字符集及其字符频率，其哈夫曼编码不一定是唯一的，但是每个字符的哈夫曼码的**长度**一定是唯一的。答案：F

提示：在某些字符频率相同的情况下，字符的长度不唯一。如在所有字符频率均相同的情况下，在不同的编码顺序下，同一个字符的长度也可能不同。