16-4

不会，放弃了，网上的也看不明白= =

17.4-3

考虑两种情况

$$1.\text{当删除一个元素，}\alpha < \frac{1}{4}\text{，}\textit{表规模不减}$$

$$2.\text{当删除一个元素，}\alpha < \frac{1}{3}\text{，}\textit{表规模减少为原来的}\frac{2}{3}$$

我们先来计算第 1 种情况的摊还代价

$$\because \text{表规模不减，所以删除一个元素的代价是 } 1$$

$$\hat{c} = \mathrm{c_i} + \phi_\mathrm{i} - \phi_{i-1} = 1 + \mathrm{T.size(i)} - 2 * \mathrm{T.num(i)} - \big(\mathrm{T.size(i-1)} - 2 * \mathrm{T.num(i-1)}\big)$$
$$\because \mathrm{T.size(i)} = \mathrm{T.size(i-1)}, \mathrm{T.num(i)} = \mathrm{T.num(i-1)} - 1$$
$$\therefore \hat{c} = 3$$

在第二种情况下

$$\mathrm{T.size(i)} = \frac{2}{3}T.size(i-1), T.num(i) = T.num(i-1) - 1$$

$$\because \text{删除一个元素后，表的规模改变，所以删除的代价为 } \mathrm{T.num(i)} + 1$$

$$\hat{c} = c_i + \phi_\mathrm{i} - \phi_{i-1} = T.num(i-1) - \frac{1}{3}T.size(i-1) + 2$$

$$\because \alpha = \frac{1}{3} = \frac{T.num(i)}{T.size(i)}$$
$$\therefore \hat{c} = 2$$

$$\therefore \text{摊还代价的上界是一个常数}$$

17-2

a.

k 个数组都是有序的，挨个儿找就行了，在每个数组上执行二分查找。

当最坏情况发生时，就是每个数组都执行了lgm次查找，而且找到最后一个数组才找到，其中 m 是每个数组的大小。

$$\therefore \mathrm{T(n)} = \Theta\left(\lg 2^{k-1} + \lg 2^{k-2} + \cdots + \lg 2^0\right) = \Theta\left(\frac{k(k-1)}{2}\right) = \Theta(\lg^2 n)$$

b.

采用一种循环的方式。

一开始新建一个大小为 1 的数组A_t，只放我们准备插入的元素。

然后看A_0是否是空的, 如果是空的, $A_0 = A_t$; 如果不是空的, 把A_0*和*A_t*的元素merge后排序,*

这样数组的大小是 2，然后看A_1是不是空的，如果是空的，则A_1=merge 后排序的数组，如果不是, 继续这个步骤, 直到找到一个空数组能装下前面所有数组的 merge 后排序的数组。

$$\because \text{大小为 a 和 b 的数组 merge} - \text{sort 的时间是 } O(a+b)$$

$$\therefore \text{在最坏情况下，} \mathrm{T(n)} = 2 * \left(2^0 + 2^1 + \cdots 2^{k-2}\right) = \Theta(n)$$

采用记账法进行均摊分析，设每次插入赚 k$，代价 1$，则净利润(k-1)$，而且后面 merge 的代价至多是（k-1）$，所以钱是够的。$\because k = \Theta(\lg n), \therefore$ 均摊分析的代价为 $\Theta(\lg n)$

c.

1. 找到下标最小的满*数组*A_j
2. 找到要删除的 x 在哪个数组中
3. 从 x 所在数组把 x 删掉，然后把A_j*的最后一个元素拿过来，排好序*
4. 把A_j*数组中的剩下*$2^j - 1$ *个元素分成*$A_0, A_1, A_2, \ldots, A_{j-1}$，令$A_j$*为空*