17.4-3

证明：

(1) $\alpha_i \leq \frac{1}{3}$

$$\begin{aligned}\hat{c}_i =& c_i + \Phi_i - \Phi_{i-1} \\ =& 1 + (2 \cdot \text{num}_i - \text{size}_i) \\ & - \left(2 \cdot \text{num}_{i-1} - \text{size}_{i-1}\right) \\ =& 1 + 2 \cdot (i - (i-1)) \\ =& 3\end{aligned}$$

(2) $\alpha_i < \frac{1}{3}, \alpha_{i-1} \geq \frac{1}{3}$

$$\begin{aligned}\hat{c}_i =& c_i + \Phi_i - \Phi_{i-1} \\ =& (num_i + 1) + (2 \cdot \text{num}_i - \text{size}_i) \\ & - \left(2 \cdot \text{num}_{i-1} - \text{size}_{i-1}\right) \\ =& (num_i + 1) + 0 - |2 \cdot (num_i + 1) - 3 \cdot num_i| \\ =& 3\end{aligned}$$

所以摊还代价的上界是常数3

17-2

(a)

算法：从n_0 开始，如果n_i 不为0，则依次遍历k个有序数组，在每个数组中进行二分查找。

最坏时间复杂度为:$1 + 2 + 3 + \cdots + \lceil lg(n+1) \rceil = \Theta(lg^2(n))$

(b)

算法：对于INSERT操作，从n_0 开始，找到第一个为0的n_i，将n_i 之前的所有数组和要插入的数合并，全部存入n_i 对应的数组中，之前的所有数组清空，n_i 设为1，$n_j(j < i)$全部设为0。

最坏时间复杂度发生在只有n_{k-1} 为0的情况下，此时需要在$\lceil lg(n+1) \rceil$ 个有序数组中合并$n/2$个数，时间复杂度为：$\Theta(n \cdot lgn)$

摊还时间分析：每次插入，n_i 从0到1翻转的可能性为$\frac{1}{2^{i+1}}$，时间花费为$\Theta(2^i \cdot (i+1))$，故总时间花费为$\Theta(\sum \frac{1}{2^{i+1}} 2^i (i+1)) = \Theta(lg^2(n))$

(c)

现将DELETE的元素a删除。然后，从n_0 开始，找到第一个为1的n_i，从中取出最小的元素，找到合适位置填入失去a的数组中。最后，将n_i 数组的元素全部转移到$n_j(j < i)$的数组中，将n_i 设为0，$n_j(j < i)$全部设为1.

19-3

(a)

$k < x.key$时，分析方法同FIB-HEAP-DECREASE-KEY，摊还时间为$\Theta(1)$

$k = x.key$时，无需做任何改变，摊还时间为$\Theta(1)$

$k > x.key$时，由于和DECREASE-KEY操作不同，其不参与DELETE操作，所以只需判断是否大于其孩子，如果大于一些孩子，则对这些孩子执行$CUT(H, x, y)$和$CASCADING - CUT(H, y)$操作，每次操作的摊还时间为$\Theta(1)$，最多不超过$lg(n)$次，故总摊还时间为$O(lg(n))$

(b)

将势函数修改为： $\Phi(H) = t(H) + 2m(H) + n$

其中n为节点数量。同时，将所有叶节点都有用双向链表进行连接。

进行裁剪时，选择任何一个叶子结点，将从父节点孩子列表中移除，也从叶子结点链表中移除。重复该操作q次，势函数也会降低n，所以摊还时间为常数$\Theta(1)$