600.363/463 Solution to Homework 4
Due: Wednesday, 10/8

17.3-1

Define $\Phi'(D_i) = \Phi(D_i) - \Phi(D_0)$ for all $i \geq 1$.
Then $\Phi'(D_0) = \Phi(D_0) - \Phi(D_0) = 0$, and $\Phi'(D_i) = \Phi(D_i) - \Phi(D_0) \geq 0$.
The amortized cost:
$\hat{c_i}' = c_i + \Phi'(D_i) - \Phi'(D_{i-1})$
$= c_i + (\Phi(D_i) - \Phi(D_0)) - (\Phi(D_{i-1}) - \Phi(D_0))$
$= c_i + \Phi(D_i) - \Phi(D_{i-1})$
$= \hat{c_i}$

17.3-4

The total costs is bounded by $(\leq)$ $2n + s_0 - s_n$.

17.4-2

If $\alpha_{i-1} \geq 1/2$, TABLE-DELETE cannot trigger contraction, so $c_i = 1$, $size_i = size_{i-1}$.
if $\alpha_i \geq 1/2$:
$\hat{c_i} = c_i + \Phi_i - \Phi_{i-1}$
$= 1 + (2 \cdot num_i - size_i) - (2 \cdot num_{i-1} - size_{i-1})$
$= 1 + (2 \cdot (num_{i-1} - 1) - size_{i-1}) - (2 \cdot num_{i-1} - size_{i-1})$
$= -1$
if $\alpha_i < 1/2$:
$\hat{c_i} = c_i + \Phi_i - \Phi_{i-1}$
$= 1 + (size_i/2 - num_i) - (2 \cdot num_{i-1} - size_{i-1})$
$= 1 + (size_{i-1}/2 - (num_{i-1} - 1)) - (2 \cdot num_{i-1} - size_{i-1})$
$= 2 + \frac{3}{2} size_{i-1} - 3 \cdot num_{i-1}$
$= 2 + \frac{3}{2} size_{i-1} - 3\alpha_{i-1} size_{i-1}$
$\leq 2 + \frac{3}{2} size_{i-1} - \frac{3}{2} size_{i-1}$
$= 2$

17.4-3

When there is no contracion, which means $\alpha_{i-1} \geq \frac{1}{3}$ and $\alpha_i \geq \frac{1}{3}$, $c_i = 1$.
a. if $\alpha_{i-1} \geq \frac{1}{2}$ and $\alpha_i \geq \frac{1}{2}$
$\hat{c_i} = c_i + \Phi_i - \Phi_{i-1}$
$= 1 + 2 \cdot num_i - size_i - (2 \cdot num_{i-1} - size_{i-1})$
$= 1 + 2(num_{i-1} - 1) - size_{i-1} - 2 \cdot num_{i-1} + size_{i-1}$
$= 1 - 2 = -1$
b. if $\alpha_{i-1} \geq \frac{1}{2}$ and $\alpha_i < \frac{1}{2}$
$\hat{c_i} = c_i + \Phi_i - \Phi_{i-1}$
$= 1 + size_i - 2 \cdot num_i - (2 \cdot num_{i-1} - size_{i-1})$
$= 1 + size_{i-1} - 2(num_{i-1} - 1) - 2 \cdot num_{i-1} + size_{i-1}$
$= 3 + 2 \cdot size_{i-1} - 4 \cdot num_{i-1}$

$= 3 + 2 \cdot size_{i-1} - 4\alpha_{i-1} size_{i-1}$
$\leq 3 + 2 \cdot size_{i-1} - 4 \cdot \frac{1}{2} size_{i-1}$
$= 3$

c. if $\frac{1}{3} \leq \alpha_{i-1} < \frac{1}{2}$ and $\frac{1}{3} \leq \alpha_i < \frac{1}{2}$
$\hat{c_i} = c_i + \Phi_i - \Phi_{i-1}$
$= 1 + size_i - 2 \cdot num_i - (size_{i-1} - 2 \cdot num_{i-1})$
$= 1 + size_{i-1} - 2(num_{i-1} - 1) - size_{i-1} + 2 \cdot num_{i-1})$
$= 3$

If $\alpha_{i-1} \geq \frac{1}{3}$, and the DELETE operation triggers a contraction, the actual cost of the operation is $c_i = num_i + 1$, $size_i = \frac{2}{3} size_{i-1}$, and $num_{i-1} = \frac{1}{3} size_{i-1}$.
$\hat{c_i} = c_i + \Phi_i - \Phi_{i-1}$
$= (num_i + 1) + size_i - 2 \cdot num_i - (size_{i-1} - 2 \cdot num_{i-1})$
$= \frac{1}{3} size_{i-1} + 1 + \frac{2}{3} size_{i-1} - 2(\frac{1}{3} size_{i-1} - 1) - size_{i-1} + 2 \cdot \frac{1}{3} size_{i-1}$
$= 1$

So it is bounded as: $\hat{c_i} \leq 3$

21.2-3

We want to show that we can assign $O(1)$ charges to MAKE-SET and FIND-SET and a $O(\lg n)$ charge to UNION such that the charges for a sequence of these operations are enough to cover the cost of the sequence $O(m + n \lg n)$, according to the theorem. When talking about the charge for each kind of operation, it is helpful to also be able to talk about the number of each kind of operation.
Consider the usual sequence of m MAKE-SET, UNION, and FIND-SET operations, n of which are MAKE-SET operations, and let $l < n$ be the number of UNION operations. ($l < n$ because there are only n sets available to combine.) Then there are n MAKE-SET operations, l UNION operations, and $m - n - l$ FIND-SET operations.
The theorem didn't seperately name the number l of UNIONs; rather it bounded the number by n. If you go through the proof of the theorem with l UNIONs, you get the time bound $O(m - l + l \lg l) = O(m + l \lg l)$ for the sequence of operations. That is, the sequence of operations takes $\leq c(m + l \lg l)$ time for some constant c.
So we want to assign operation charges such that

(MAKE-SET charge)$\cdot n$ +(FIND-SET charge)$\cdot(m - n - l)$ +(UNION charge)$\cdot l$ $\geq c(m + l \lg l)$.

The following assignments work, where c' is some constant $\geq c$:
MAKE-SET: c'
FIND-SET: c'
UION: $c'(\lg n + 1)$
Substituting into the above sum we get:
$c'n + c'(m - n - l) + c'(\lg n + 1)l = c'm + c'l \lg n$
$= c'(m + l \lg n)$
$> c(m + l \lg l)$