1. Suppose we perform a sequence of stack operations on a stack whose size never exceeds $k$. After every $k$ operations, we make a copy of the entire stack for backup purposes. Show that the cost of $n$ stack operations, including copying the stack, is $O(n)$ by assigning suitable amortized costs to the various stack operations.

2. Suppose we have a potential function $\Phi$ such that $\Phi(D_i) \geq \Phi(D_0)$ for all $i$, but $\Phi(D_0) \neq 0$. Show that there exists a potential function $\Phi'$ such that $\Phi'(D_0) = 0$, $\Phi'(D_i) \geq 0$ for all $i \geq 1$, and the amortized costs using $\Phi'$ are the same as the amortized costs using $\Phi$.

3. What is the total cost of executing $n$ of the stack operations Push, Pop, and Multipop, assuming that the stack begins with $s_0$ objects and finishes with $s_n$ objects?

4. Show that if $\alpha_{i-1} \geq 1/2$ and the $i$th operation on a dynamic table is Table-Delete, then the amortized cost of the operation with respect to the potential function

$$\Phi(T) = \begin{cases} 
2 \cdot T.num - T.size & \text{if } \alpha(T) \geq 1/2, \\
T.size/2 - T.num & \text{if } \alpha(T) < 1/2.
\end{cases}$$

is bounded above by a constant.

5. Suppose that instead of contracting a table by halving its size when its load factor drops below $1/4$, we contract it by multiplying its size by $2/3$ when its load factor drops below $1/3$. Using the potential function

$$\Phi(T) = |2 \cdot T.num - T.size|$$

show that the amortized cost of a Table-Delete that uses this strategy is bounded above by a constant.

This is the last set of exercise questions for 2013.