ALGORITHMS, FALL 2018, HOMEWORK 9

1. (a) To maximize the reward per move, you need to first spend some moves setting up the correct stacks of bricks. If you take the first $k - \sqrt{k}$ bricks, and redistribute their contents to the last $\sqrt{k}$ bricks, you are left with $\sqrt{k}$ stacks with each stack having exactly $\sqrt{k}$ bricks. This strategy will maximize the average reward to $\sqrt{k}$, because now all of the remaining moves will simply be redistributing each stack among the other existing stacks, plus one empty stack. This allows us to keep the reward constant at $\sqrt{k}$.

   (b) To use potential method to amortize this strategy, we can say that the maximum potential score for each move is $\sqrt{k}$.

   This means that c_i is $\sqrt{k}$.

   Now any move that we make that is greater than $\sqrt{k}$ will have its excess banked for later use.

   If we define our potential function as $c_i + \Delta\Phi_i$, where $\Phi_1 - \Phi_0 = h - (h - \sqrt{k})$, we're left with $2\sqrt{k}$ as our upper bound for potential.

   Now we have to prove that $\sum c_i \leq \sum n\hat{c}_i$, which is equivalent to $n * \sqrt{k} \leq n * 2\sqrt{k}$.

   This is basically $1 \leq 2$.

(c) For the accounting method, if we pretend that the $k - \sqrt{k}$ operations take $\sqrt{k}$, the deficit becomes $k\sqrt{k} - k$. We only need to support $k - \sqrt{k} * \sqrt{k} - 1$ operations to bank back the stuff we lost from the initial setup.

This makes the lower bound on the score still $\sqrt{k}$.

2. For this problem, let us collect all the roads in an array. Every road only needs to know what kind of city it connects to. This allows us to iterate over all E of the roads.

   Then let us assume that boring cities don't exist (because they do not matter in the context of this problem). Therefore we can iterate all of the V cities, and if the city is boring, go through all of it's roads and create new connections (road) between every city it connects. The maximum time complexity for this is $E * V$, because if every city is boring the problem is finished.

   This means that we can mark all boring cities surrouneded by awful cities as also awful, and then all awful cities that are not surrounded by awful cities as visitable. The runtime for this is VE.