Avi Hirsch
Design and Analysis of Algorithms
HW #8: KT 4.3, 4.5

4.3. *Prove that, for a given set of boxes with specified weights, the greedy algorithm currently in use actually minimizes the number of trucks that are needed.*

Given a set of boxes $b_1$ to $b_n$, where each box $b_i$ has a weight $w_i$ and the maximum weight a truck can carry is W, to prove that the greedy algorithm is the optimal algorithm, we must show that the current greedy algorithm they use is only as good or better (never worse) than the "optimal" algorithm. In other words, if the greedy algorithm can fit n boxes on the first k trucks, and the optimal algorithm can fit m boxes on the same number of trucks, we must prove that
$n \geq m$; i.e., that the greedy algorithm can fit as many or more boxes on the same number of trucks as the optimal algorithm.

To prove this, consider the first truck. The greedy algorithm puts as many boxes as it can on that truck, so it must have as many or more on it than the optimal algorithm. Now consider the next k trucks. For each truck, the greedy algorithm will put as many boxes as can fit on that truck; the optimal algorithm can at best do the same. Thus, for the k + 1 truck, the greedy algorithm will fill it up to its capacity, say, with i boxes, and the optimal algorithm will fill it with j boxes; now, because i is the most number of boxes that can fit on the truck, i must be greater than or equal to j. Thus, the greedy algorithm will always be as good or better than the optimal algorithm.

4.5. *You want to place cell phone base stations at certain points along the road, so that every house is within four miles of one of the base stations. Give an efficient algorithm that achieves this goal, using as few base stations as possible.*

To use as few stations as possible, it is necessary to place them at intervals such that the most houses are covered with the fewest stations. Moving from one end of the road to the other (either west -> east or vice-versa), place a station $s_0$ as far from the starting end (which we will call position $p_0$) as possible such that you haven't missed a house. In other words, the house closest to $p_0$ should be 4 miles closer to $p_0$ than $s_0$. Remove all houses covered by $s_0$, and proceed in the same direction. Repeat until you reach the opposite end.

This greedy algorithm is the most optimal, which can be shown in the same way as we proved in question 4.3. Comparing the greedy algorithm to the optimal algorithm, we see that moving from starting end $p_0$, the optimal algorithm will place its first station $t_0$ no further away from $p_0$ than the greedy algorithm's first station $s_0$, because if it was any further away, it would not cover the first house. So for the first house, the greedy algorithm is as good as the optimal algorithm.

Now, assume the greedy algorithm stays ahead of the optimal algorithm for n stations. Then, the greedy algorithm will place its $n+1^{th}$ station $s_{n+1}$ as far as possible such that no house is missed; the optimal algorithm cannot do any better than that. Thus, its station $t_{n+1}$ cannot be any further than $s_{n+1}$, or $s_{n+1} \geq t_{n+1}$. We have thus proven that the greedy algorithm is optimal.