2.

a) The Truck Loading problem is similar to the Load Balancing problem in Section 11.1 of the textbook .The algorithm provided takes the most recent weight and adds it to the earliest unfilled truck. We know each truck has capacity K pounds.

To show that the algorithm provided is 2-optimal, we must show that it never uses twice as many trucks as needed. Then, we must show that if m\* is the optimal number of trucks for the problem, the algorithm provided uses less than 2m\* trucks.

A natural lower bound for the optimal number of trucks . Intuitively, if the sum of the weights exceeds cK for some constant c, then any allocation of weights to trucks will require c+1 trucks to accommodate all weights. We can prove this by contradiction. Assume that there were a way to allocate m>cK weights in at most c trucks. Then, there is one truck with weight at least m/c. This would violate the assumption that a truck has a limit of K weight, because m/c>K. Now, we want to show:

Next, we want to place an upper bound on the number of trucks produced by the given Truck Loading algorithm. Say there are n items with weights w1, w2,… wn known a priori. We start by considering the first truck. Say weights w1, w2,… wk fit onto the truck, using the given Truck Loading algorithm. Then, the next weight is weight wk+1, which will require at most one truck. If this isn’t true, a contradiction is made because it means wk+1>K. In general, each weight considered adds at most one truck, so any two consecutive trucks in the greedy algorithm have total weight less than 2K and greater than K. The optimal allocation of weights for the subset of weights considered also uses at most 2 trucks. It follows that if any subset of weights in the greedy pair of trucks differs by at most 1 per truck, then for each truck in m\*, at most 1 truck exists in the greedy allocation: . Thus, the proof is complete.

b) Say each truck has capacity K>>1 . Consider a sequence of (finite) considered items of the following pattern: {K, 1, K, 1…} or {1, K, 1, …} . We regard the addition of each consecutive, non-overlapping pair of items to begin with, and assume there are n total items for some even number n>2.

Then, the first pair has m\*=ceil(K+1/K) = 2 trucks and the greedy algorithm also uses 2 trucks, one for each item.

Next, adding another item of weight K and another item of weight 1, , while the greedy algorithm uses 4 trucks, one per item.

From this we can form an inductive hypothesis: the ith weight (not pair) will have m\* = ceil(i/2). In the greedy case, we will always use i trucks. This holds for any weight considered for OPT by the formula we have defined above. Similarly, we know that our greedy algorithm must use i trucks for the ith weight of this sequence because having more fewer than i trucks means some truck has weight over K and having more than i trucks means an item weight exceeded K or at least one truck is empty, all of which present contradictions. As a result, this inductive hypothesis is valid for the ith weight of this particular sequence presented. As a result, malg=2m or 2m-1, both of which are greater than 2m\*-2. Thus, the upper-bound defined is tight.