Problem A.5: Writeup

1. What were your results from compare\_cow\_transport\_algorithms? Which algorithm runs faster? Why?

*The “greedy algorithm” runs faster, 0.0000451 s vs. 0.58 s used by the “brute force” algorithm. The greedy algorithm implements a logic that reduces the search space of the problem.*

1. Does the greedy algorithm return the optimal solution? Why/why not?

*It does not. Although it shrinks the search space, we can only say that it finds a sub-optimal solution.*

1. Does the brute force algorithm return the optimal solution? Why/why not?

*It does, because it explores for the entire search space scanning local minima until it finds a global optimal solution.*

Problem B.2: Writeup

1. Explain why it would be difficult to use a brute force algorithm to solve this problem if there were 30 different egg weights. You do not need to implement a brute force algorithm in order to answer this.

*We would need to generate all set partitions for n=30, which has an exponential time complexity of O(2^n). Of course, we could filter out those partitions that do not fit the total weight constraint, but the search complexity would still be the same.*

1. If you were to implement a greedy algorithm for finding the minimum number of eggs needed, what would the objective function be? What would the constraints be? What strategy would your greedy algorithm follow to pick which eggs to take? You do not need to implement a greedy algorithm in order to answer this.

*The objective function would be the total number of eggs taken and the only constraint would be the total weight available in the ship.*

*My greedy algorithm:*

1. *Define a score, , where is the max number of eggs of weight “i” that fit in the weight available and is the remaining weight after fitting these number of eggs. “W” is the weight available in the ship after a number of eggs is taken*
2. *We start off with the total ship weight and compute the score “s” for every egg weight.*
3. *Pick the weight with the lowest score, take as many eggs into the ship and compute the weight left “W”*
4. *Stop when W=0*
5. Will a greedy algorithm always return the optimal solution to this problem? Explain why it is optimal or give an example of when it will not return the optimal solution. Again, you do not need to implement a greedy algorithm in order to answer this.

It will not. The greedy algorithm might start fitting in the ship with the heaviest eggs and then being left with a small weight that can only be filled with a bunch of lighter eggs. This decision could lead to carrying more eggs than necessary.