Problem A

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

Trips searched by Greedy alogorithm is [['Betsy'], ['Henrietta'], ['Herman', 'Moo Moo'], ['Oreo', 'Maggie'], ['Millie', 'Lola', 'Florence'], ['Milkshake']]

cost time = 0.000997304916381836s.

Trips searched by Greedy alogorithm is [['Millie', 'Milkshake', 'Maggie'], ['Henrietta'], ['Moo Moo', 'Herman'], ['Oreo', 'Lola', 'Florence'], ['Betsy']]

cost time = 1.4641788005828857s.

We can easily see that the greedy algorithm is much faster than brute force algorithm.

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

No, it does not return the optimal solution because it does not find the best solution containing most single trip whose weight is equal to weight limit.

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

Yes, it returns the optimal solution because this one is the minimum among all the possible trips.

Problem B

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.

Because there are so many possibilities for 30 different eggs, and it requires a lot of time to implement this algorithm.

2. 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 coins to take? You do not need to implement a greedy algorithm in order to answer this.

Objective function: Find the minimum number of eggs with different weights needed for a target weight.

Constraints: the maximal weight or target weight, the kinds of eggs with different weights.

Strategy: find the maximal number of heaviest eggs satisfying that the total weight is equal or less than and most close to target weight. And then calculate the remaining weight and find the maximal number of second heaviest eggs satisfying that total weight is equal or less than and most close to target weight. Doing in this way, until the left weight is zero. And then the total number of different eggs would be the minimum number.

3. 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.

Yes, it will in this case. This is different with the calculating way in problem A. Because this is one-time trip, every time the program will choose the egg which make the total weight of all chosen eggs is equal or less than and most close to target weight. There is no other satisfied number smaller than this minimum found by greedy algorithm.