Problem A.5: Writeup

1. Results from compare_cow_transport_algorithms:
    For greedy algorithm, time taken = 3.0994415283203125e-05 s and no. of trips = 6
    For brute force algorithm, time taken = 0.5800626277923584 s and no. of trips = 5
We see that the greedy algorithm runs much faster than the brute force algorithm since the greedy algorithm is O(n^2) where n is the number of cows whereas the brute force algorithm is exponential w.r.t. the number of cows.

2. The greedy algorithm doesn't return the optimal solution since it makes greedy decisions i.e. it makes locally optimal choices which does not necessarily result in a global optimal choice.

3. The brute force algorithm does return the optimal solution since it checks all the possible combinations of trips and chooses the best out of them.

Problem B.2: Writeup

1. It would be very difficult to use a brute force algorithm to solve this problem if there were 30 different egg weights since that were to mean that each node/weight could have 30 different children so the tree would get imensely large and too expensive to compute because we would be computing the same subproblems repeatedly even if we had already computed it before.

2. If I were to implement a greedy algortihm to find the minimum number of eggs needed, the objective function would be the number of eggs which is to be minimised.
The constraints would be that an egg will be taken only if it's weight is less than or equal to the available space in the spaceship.
The strategy my greedy algorithm will follow to pick which eggs to take would be to take as many eggs of max weight first which the spaceship can accomodate and then take as many eggs from the next available max weight which the spaceship can accomodate and keep going on to selecting the eggs of next available max weights which the spaceship can accomodate till the target weight is not reached.

3. No, the greedy algorithm will not always return the optimal solution to the problem since it will always make local choices of optimization which may not necessarily lead to the best global optimal choice.
Example when it would not give an optimal answer:
Egg weights = (1, 3, 4)
Target weight = 6
Greedy choice = 3 eggs i.e. of weights (4, 1, 1)
Actual choice = 2 eggs i.e. of weights (3, 3)