Problem Set 1A Answers

John L. Jones IV

December 10, 2019

1

What were your results from `compare_cow_transport_algorithms`? Which algorithm runs faster? Why?

Using `ps1_cow_data.txt` the results were:

```
greedy_cow_transport:
length = 6 trips
time = .000145 seconds

brute_force_cow_transport:
length = 5 trips
time = 0.48294 seconds
```

The algorithm `greedy_cow_transport` does not iterate through every possible combination of trips like `brute_force_cow_transport`. In my implementation of `greedy_cow_transport`, first the input dictionary of cows is copied to a list of cow names `sorted` from largest to smallest weight. The python `sorted` function utilizes a Timsort which is $\mathcal{O}(n \log n)$. Then `greedy_cow_transport` removes any cow which is larger than `limit`, an $\mathcal{O}(n)$ operation. The sorted list is then utilize to select the cows which can fit on the ship. Starting at the big end of the list, iterate and select cows that can fit onto the ship without exceeding the payload `limit`. Since the list has been sorted, this is an $\mathcal{O}(n \log n)$ operation. The python `sorted` function's Timsort and selecting cows operation dominate the run time, making `greedy_cow_transport` $\mathcal{O}(n \log n)$. In comparison, `brute_force_cow_transport` must first create all permutations of the possible trips, $\mathcal{O}(n^2)$. Then evaluate each of these trips, $\mathcal{O}(n^2)$. This emphasizes Professor John Guttag's quote from Lecture 1, "many optimization problems are inherently exponential. What that means is there is no algorithm that provides an exact solution to this problem whose worst case running time is not exponential in the number of items."

2

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

The greedy algorithm *does not* return the optimal solution. The nature of this "knapsack" problem can only be solved in $\mathcal{O}(n^2)$. However, a reasonable solution can be solve in much less time. With `ps1_cow_data.txt` and this test set the solution is solved on the order of 1/1000 of the time of brute force algorithm.

3

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

Yes, the brute force algorithm does fine the optimal solution. All possible solutions are evaluation and the best is chosen. However, this comes at a great cost to run-time speed. With `ps1_cow_data.txt` and this test set the solution is solved on the order of 1000 times slower than that of the greedy algorithm.