Assignment 3

Erik Norlin, CID: norliner

September 29, 2023

Problem 6

a)

A greedy algorithm that runs in $O(nlogn)$ is to start sorting the items by price in descending order. Then divide the set into triples starting from the beginning of the set. If the number of items is not divisible by 3, the last subset will contain either 1 or 2 items.

Pseudo code for this algorithm

```
Initialize a set A with all items
Sort A by price in descending order
Initialize an empty set F to contain all triples

While there are three or more items in A

    Make a triple out the first three items in the set A
    Add the triple to F
    Delete the first three items in A

    If there are two or less items left in A
        Make an incomplete triple out the remaining items in A
        Add the incomplete triple to F
```

The sorting runs in $O(nlogn)$, and the loop runs in $O(n)$ so the resulting running time for this algorithm is $(nlogn + n) \in O(nlogn)$.

b)

We prove that this algorithm is the most optimal one by exchange argument. Let A be the set of all items and be sorted by price in descending order, F be a compatible set of items we get for free that is returned by the greedy algorithm also sorted by price in descending order, and O be some optimal compatible set of items we get for free sorted by price in descending order as well. Additionally, let i be items of F, j be items of O, and p be the price of items. By the rules of the greedy algorithm, the two first items in A must be bought. It becomes clear that the most expensive item that we can get for free is the third item in A which the greedy algorithm guarantees. Comparing the price of the third item in A (item i_1 in F), with the most expensive item in O (item j_1), we must have that $p(i_1) \geq p(j_1)$. If we exchange $p(j_1)$ to $p(i_1)$ in O we have therefore not decreased the total maximum value we get for free. Because of this swap, we can certainly exchange the next item in O with the next item in F without breaking the rule of the campaign, and without decreasing the maximum total value of O. Repeating this for all items in O, it becomes the greedy solution without having decreased the total maximum value of it. Thus, the greedy solution is an optimal solution.