

CSCI3070U: Analysis & Design of Algorithms

Greedy Algorithms

Syed Naqvi
100590852

November 11, 2023

A)

Fractional solution for knapsack with maximum weight 25 and provided items:

```
s/fractionalKnapsack$ ./output
Maximal value in the knapsack for the given items: $75.00.

Item1: $20.00/5.00w   Percentage Included: 1.00
Item2: $45.00/15.00w  Percentage Included: 1.00
Item4: $12.00/6.00w   Percentage Included: 0.83
Item3: $15.00/10.00w  Percentage Included: 0.00
```

B)

Using Huffman's algorithm to find the encodings for the provided characters and frequencies.
Determining the file length for Huffman codes and fixed codes.

```
... 1. Character and associated huffman encoding:
    {'B': '00', 'E': '01', 'A': '100', 'H': '101', 'C': '1100', 'F': '1101', 'D': '111'}

    2. Length of a file using huffman encodings: 109

    3. Length of a file using huffman encodings: 120
```

C)

See full implementation using C++:

```
huffmans.ipynb  GreedyAlgorithms.tex  C main.cpp x
algo_analysis > greedyAlgorithmsActivities > fractionalKnapsack > C main.cpp > main()
1  #include <iostream>
2  #include "knapsack.h"
3
4  int main() {
5
6
7      // Part A: adding items of different values and weights to a knapsack
8      // of maximum weight = 25
9      int weight = 25;
10
11     // creating the list of items
12     std::vector<Item> items;
13     // item 1
14     items.push_back(Item(20, 5));
15     // item 2
16     items.push_back(Item(45, 15));
17     // item 3
18     items.push_back(Item(15, 10));
19     // item 4
20     items.push_back(Item(12, 6));
21
22     // creating the knapsack
23     Knapsack myKnapsack(weight);
24
25     myKnapsack.fractionalInsertion(items);
26     myKnapsack.print();
27
28
29
30 }
```