K. K. Wagh Institute of Engineering Education and Research, Nashik.
Department of Computer Engineering
Academic Year 2022-23

Name: Ahire Kalpesh Bapurao	**Class:** BE
Roll No.: 12	**Div:** A
Course: Laboratory Practice III	**Course Code:** 410246

Assignment No: 03

Problem Statement:

Write a program to solve a fractional Knapsack problem using a greedy method.

Objective:

To implement Fractional Knapsack as well as comprehend greedy strategy.

Course Outcome:

CO5: Implement an algorithm that follows one of the following algorithm design strategies: divide and conquer, greedy, dynamic programming, backtracking, branch andbound.

Overview of Fractional Knapsack:

- Given the weights and values of N items, in the form of {value, weight} put these items in a knapsack of capacity W to get the maximum total value in the knapsack. In Fractional Knapsack, we can break items for maximizing the total value of the knapsack
- This problem can be solved with the help of using two techniques:
  - Brute-force approach: The brute-force approach tries all the possible solutions with all the different fractions but it is a time-consuming approach.
  - Greedy approach: In Greedy approach, we calculate the ratio of profit/weight, and accordingly, we will select the item. The item with the highest ratio would be selected first.
- There are basically three approaches to solve the problem:
  - The first approach is to select the item based on the maximum profit.
  - The second approach is to select the item based on the minimum weight.
  - The third approach is to calculate the ratio of profit/weight.

Time complexity:

- Time Complexity: O(N * log N)
- Auxiliary Space: O(N)

Steps to build Fractional Knapsack

- Calculate the ratio(value/weight) for each item.
- Sort all the items in decreasing order of the ratio.
- Initialize res =0, curr_cap = given_cap.
- Do the following for every item "i" in the sorted order:
    - If the weight of the current item is less than or equal to the remaining capacity then add the value of that item into the result
    - Else add the current item as much as we can and break out of the loop.
- Return res.

Program:

```
"""
Created on Sat Oct 22 13:46:40 2022

@author: Bardiya
Name: Asmeeta Lalit Bardiya
Class:BE-A- Computer
Lab Assignment No: 3
Title: Write a program to solve a fractional Knapsack problem using a greedy method.
"""



# Structure for an item which stores weight and
# corresponding value of Item
class Item:
    def __init__(self, value, weight):
        self.value = value
        self.weight = weight

# Main greedy function to solve problem
def fractionalKnapsack(W, arr):

    # Sorting Item on basis of ratio
    arr.sort(key=lambda x: (x.value/x.weight), reverse=True)

    # Result(value in Knapsack)
    finalvalue = 0.0

    # Looping through all Items
    for item in arr:

        # If adding Item won't overflow,
        # add it completely
        if item.weight <= W:
            W -= item.weight
            finalvalue += item.value
        # If we can't add current Item,
        # add fractional part of it
        else:
            finalvalue += item.value * W / item.weight
            break
    # Returning final value

return finalvalue

# Driver Code
if __name__ == "__main__":

    W = 50
    arr = [Item(60, 10), Item(100, 20), Item(120, 30)]

    # Function call
    max_val = fractionalKnapsack(W, arr)
    print("Maximum Value of Knapsack", max_val)
```

Output:

```
Maximum Value of Knapsack 240.0
```

Conclusion:

Thus, I learnt the concept of greedy strategy as well as implemented algorithm to solve fractional knapsack problem using this strategy.