



Artificial Intelligence and Data Science Department.

AOA / Even Sem 2021-22 / Experiment 3.

YASH SARANG.

47 / D6AD.

EXPERIMENT - 3.

Aim: Write a C program to implement fractional Knapsack Problem.

Theory:

Given weights and values of n items, we need to put these items in a knapsack of capacity W to get the maximum total value in the knapsack.

In the 0-1 Knapsack problem, we are not allowed to break items. We either take the whole item or don't take it.

In Fractional Knapsack, we can break items for maximizing the total value of knapsack. This problem in which we can break an item is also called the fractional knapsack problem.

A brute-force solution would be to try all possible subset with all different fractions but that will be too much time taking.

An efficient solution is to use Greedy approach. The basic idea of the greedy approach is to calculate the ratio value/weight for each item and sort the item on the basis of this ratio. Then take the item with the highest ratio and add them until we can't add the next item as a whole and at the end add the next item as much as we can. Which will always be the optimal solution to this problem.

Time Complexity: $O(n)$

Time Complexity of fractional knapsack would be n , since we just have to take the ratio of the value by weight and input according to the increasing order of that ratio.

CODE:

Code is in the fractional_knapsack.c file attached along with this doc.

OUTPUT:

```
Added object 5 (10$, 4Kg) completely in the bag. Space left: 11.
Added object 2 (2$, 1Kg) completely in the bag. Space left: 10.
Added object 3 (2$, 2Kg) completely in the bag. Space left: 8.
Added object 4 (1$, 1Kg) completely in the bag. Space left: 7.
Added 58% (4$, 12Kg) of object 1 in the bag.
Filled the bag with objects worth 17.33$.
```

CONCLUSION:

By performing this experiment,

I can conclude that as compared to the 1/0 knapsack problem, the fractional knapsack problem is easier to solve and understand with a straight approach of inserting with respect to the value by weight ratio.
