Practical No. 3

Title:

Write a Program to Solve Fractional Knapsack Problem Using a Greedy Method

Aim:

To implement a program to solve the Fractional Knapsack problem using the greedy algorithm and determine the maximum profit achievable.

Objective:

To apply the greedy strategy by selecting items with the highest value-to-weight ratio first, allowing fractional amounts of items to maximize total profit.

Hardware Requirements:

- Processor: Intel Core i3 or higher
- RAM: Minimum 2 GB
- Storage: Minimum 100 MB free space
- Input Devices: Keyboard and Mouse
- Output Device: Monitor

Software Requirements:

- Operating System: Ubantu/Windows
- Programming Language: Python 3.x (or C/C++/Java)
- IDE/Text Editor: VS Code / PyCharm / Notepad++
- Terminal or Command Prompt

Theory:

The Fractional Knapsack problem is a classic optimization problem where:

- You have a knapsack with a weight capacity W.
- There are n items each with weight w[i] and profit/value p[i].
- Unlike the 0/1 knapsack problem, you can take fractions of items.
- The goal is to maximize the total profit without exceeding the capacity.

The greedy approach sorts items by their value-to-weight ratio (p[i]/w[i]) and picks the most profitable fraction first until the knapsack is full.

Algorithm (Greedy Method):

1. Calculate value/weight ratio for each item.
2. Sort all items in descending order of their ratio.
3. Initialize total profit = 0 and remaining capacity = W.
4. For each item in sorted order:
    - If the item can fit completely, add full profit and reduce capacity.
    - Else, add fractional profit proportional to remaining capacity and fill knapsack.
5. Return total profit.

Conclusion:

The fractional knapsack problem is optimally solved by a greedy approach that prioritizes items based on their value-to-weight ratio. This approach ensures maximum profit and runs efficiently in O(n log n) due to sorting.