EXPT 7(b): Knapsack problem using Greedy Method

Name: Ganesh K

USN: 2VX23UE015

In [5]:

```python
weight = [10, 20, 30]
profit = [60, 100, 120]
capacity = 40
n = len(profit)


def greedy_knapsack(capacity, weight, profit, n):

    ratio = [(profit[i] / weight[i], weight[i], profit[i]) for i in range(n)]

    ratio.sort(reverse=True, key=lambda x: x[0])

    total_profit = 0
    current_weight = 0

    for ratio_value, w, p in ratio:
        if current_weight + w <= capacity:
            total_profit = total_profit + p
            current_weight = current_weight + w

        else:
            remaining_capacity = capacity - current_weight
            total_profit = total_profit + ratio_value * remaining_capacity
            break

    return total_profit

max_profit = greedy_knapsack(capacity, weight, profit, n)

print("Maximum Profit Earned (Greedy Approach) = ", max_profit)
```

```
Maximum Profit Earned (Greedy Approach) = 200.0
```

In [7]:

```python
# Taking inputs form user
n = int(input("Enter the number of items : "))

 print("Enter the weight and profit for each item:")
weight = []
profit = []
 for i in range(n):
     w = int(input("Weigth of item {}: ". format(i+1)))
     p = int(input("Profit of item {}: ". format(i+1)))
     weight.append(w)
     profit.append(p)

capacity = int(input("Enter the capacity of the Knapsack: "))

def greedy_knapsack(capacity, weight, profit, n):

     ratio = [(profit[i] / weight[i], weight[i], profit[i]) for i in range(n)]

     ratio.sort(reverse=True, key=lambda x: x[0])

     total_profit = 0
     current_weight = 0

     for ratio_value, w, p in ratio:
         if current_weight + w <= capacity:
             total_profit = total_profit + p
             current_weight = current_weight + w

         else:
             remaining_capacity = capacity - current_weight
             total_profit = total_profit + ratio_value * remaining_capacity
             break

     return total_profit

max_profit = greedy_knapsack(capacity, weight, profit, n)

print("Maximum Profit Earned (Greedy Approach) = ", max_profit)
```

```
Enter the number of items : 5
Enter the weight and profit for each item:
Weigth of item 1: 2
Profit of item 1: 20
Weigth of item 2: 3
Profit of item 2: 60
Weigth of item 3: 5
```

```
Profit of item 3: 50
Weigth of item 4: 8
Profit of item 4: 70
Weigth of item 5: 9
Profit of item 5: 80
Enter the capacity of the Knapsack: 40
Maximum Profit Earned (Greedy Approach) = 280
```