

EXPT 7(a): Knapsack problem using Dynamic Programming Method ¶

Name: Ganesh K

USN: 2VX23UE015

```
In [11]: 1 # Input
2 weight = [1, 2, 3, 5, 7, 9]
3 profit = [60, 100, 120, 200, 300, 500]
4 capacity = 40
5 n = len(profit)
6
7 def DP_KnapSack(capacity, wieght, profit, n):
8     K = [[0 for x in range(capacity + 1)] for x in range(n + 1)]
9
10    for i in range(n + 1):
11        for w in range(capacity + 1):
12            if i == 0 or w == 0:
13                K[i][w] = 0
14            elif weight[i - 1] <= w:
15                K[i][w] = max(profit[i - 1] + K[i - 1][w - weight[i - 1]], K[i - 1][w])
16            else:
17                K[i][w] = K[i - 1][w]
18
19    return K[n][capacity]
20
21 max_profit = DP_KnapSack(capacity, weight, profit, n)
22
23 print("Maximum Profit Earned =", max_profit)
```

Maximum Profit Earned = 1280

```

In [1]: 1 # Taking inputs form user
        2 n = int(input("Enter the number of items : "))
        3
        4 print("Enter the weight and profit for each item:")
        5 weight = []
        6 profit = []
        7 for i in range(n):
        8     w = int(input("Weigth of item {}: ". format(i+1)))
        9     p = int(input("Profit of item {}: ". format(i+1)))
       10     weight.append(w)
       11     profit.append(p)
       12
       13 capacity = int(input("Enter the capacity of the Knapsack: "))
       14
       15 def DP_KnapSack(capacity, wieght, profit, n):
       16     K = [[0 for x in range (capacity +1)] for x in range (n+1)]
       17
       18     for i in range(n+1):
       19         for w in range (capacity + 1):
       20             if i == 0 or w == 0:
       21                 K[i][w] = 0
       22             elif weight[i-1] <= w:
       23                 K[i][w] = max(profit[i-1] + K[i-1][w-weight[i-1]], K[i-1]
       24             else:
       25                 K[i][w] = K[i-1][w]
       26
       27     return K[n][capacity]
       28
       29 max_profit = DP_KnapSack(capacity, weight, profit, n)
       30
       31 print("Maximum Profit Earned =", max_profit)

```

```

Enter the number of items : 7
Enter the weight and profit for each item:
Weigth of item 1: 2
Profit of item 1: 10
Weigth of item 2: 3
Profit of item 2: 5
Weigth of item 3: 5
Profit of item 3: 15
Weigth of item 4: 7
Profit of item 4: 7
Weigth of item 5: 1

```

```
Profit of item 5: 6
Weight of item 6: 4
Profit of item 6: 18
Weight of item 7: 3
Profit of item 7: 1
Enter the capacity of the Knapsack: 15
Maximum Profit Earned = 54
```