# Design Analysis and Algorithem, Lab Assignment 5

Bhuvana Kanakam - SE21UCSE035

October 19, 2023

## Question

**Assignment:** Implementation of 0/1 Knapsack problem using dynamic programming approach

## Answer

Below is the Python code :

```
import random
import matplotlib.pyplot as plt

def knapsack_01(weights, values, capacity):
    n = len(values)
    dp = [[0 for _ in range(capacity + 1)] for _ in range(n + 1)]

    for i in range(1, n + 1):
        for w in range(1, capacity + 1):
            if weights[i - 1] <= w:
                dp[i][w] = max(values[i - 1] + dp[i - 1][w - weights[i - 1]], dp[i - 1][
            else:
                dp[i][w] = dp[i - 1][w]

    return dp[n][capacity]

    items_10 = [(random.randint(1, 50), random.randint(10, 100)) for _ in range(10)]
    items_20 = [(random.randint(1, 50), random.randint(10, 100)) for _ in range(20)]
    items_30 = [(random.randint(1, 50), random.randint(10, 100)) for _ in range(30)]

    max_capacity_10 = sum(item[0] for item in items_10)
    max_capacity_20 = sum(item[0] for item in items_20)
    max_capacity_30 = sum(item[0] for item in items_30)

    n_values = [10, 20, 30]
    items_list = [items_10, items_20, items_30]
    max_capacity_list = [max_capacity_10, max_capacity_20, max_capacity_30]

    plt.figure(figsize=(18, 5))
    for i in range(3):
        plt.subplot(1, 3, i + 1)
        weights = [item[0] for item in items_list[i]]
        profits = [item[1] for item in items_list[i]]
        n = n_values[i]

    plt.plot(range(1, len(weights) + 1), weights, marker='o', linestyle='-', color='b',
    plt.plot(range(1, len(profits) + 1), profits, marker='o', linestyle='-', color='r',
    plt.title(f'Weights and Profits for n = {n}')
```

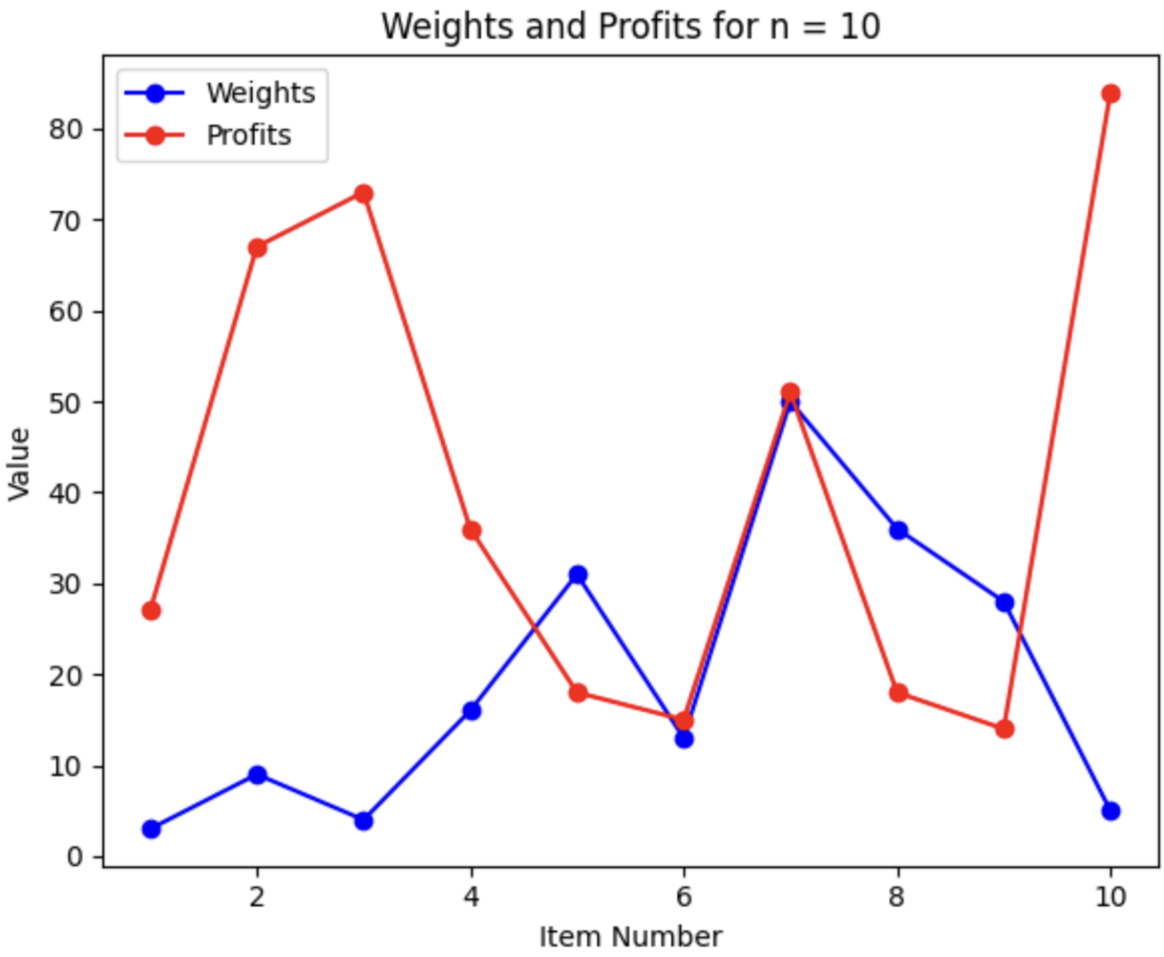

Figure 1: value for 10

```
plt.xlabel('Item Number')
plt.ylabel('Value')
plt.legend()

optimal_solutions = [knapsack_01([item[0] for item in items], [item[1] for item in i
print(f"Optimal Solutions for n=10, n=20, n=30: {optimal_solutions}")
```

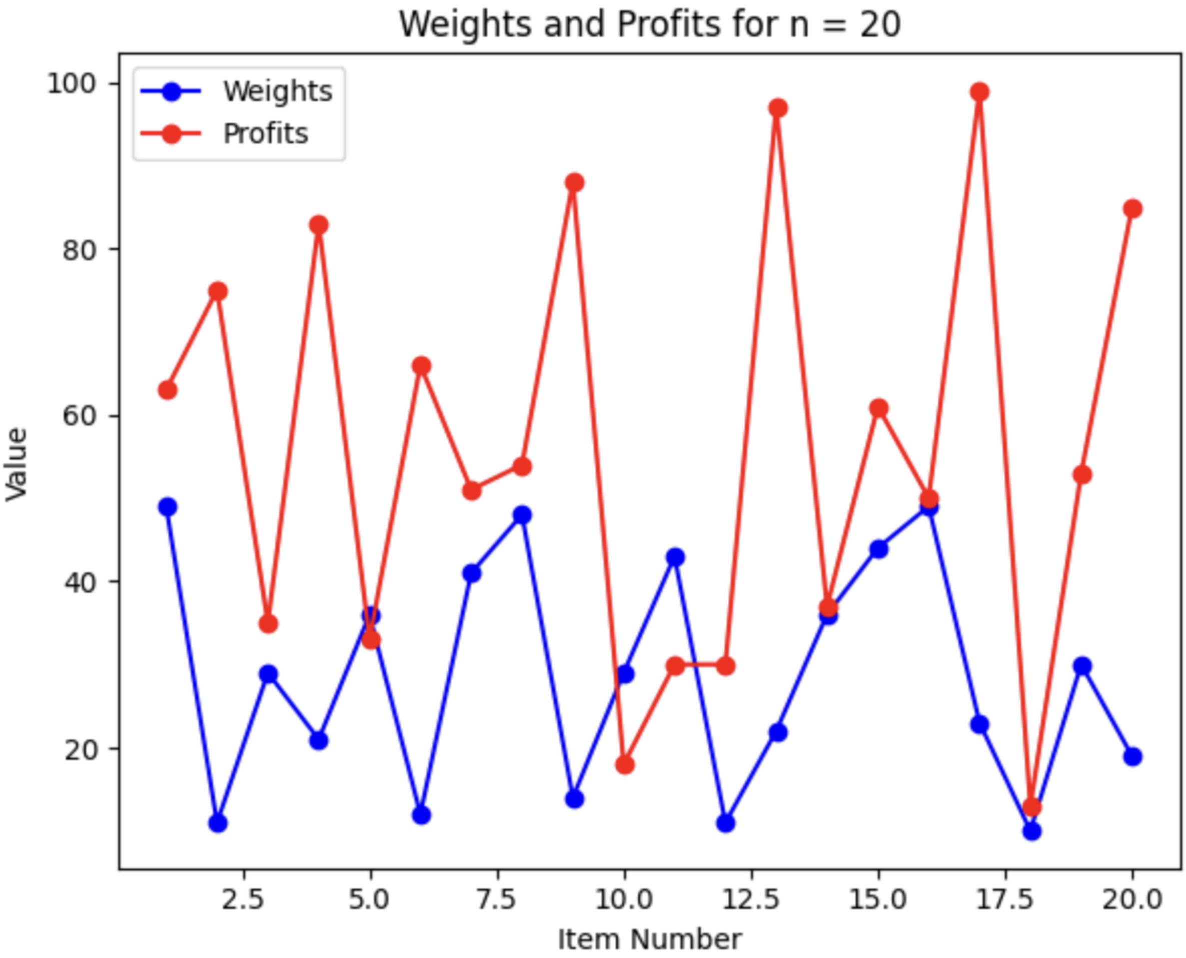

Figure 2: value for 20

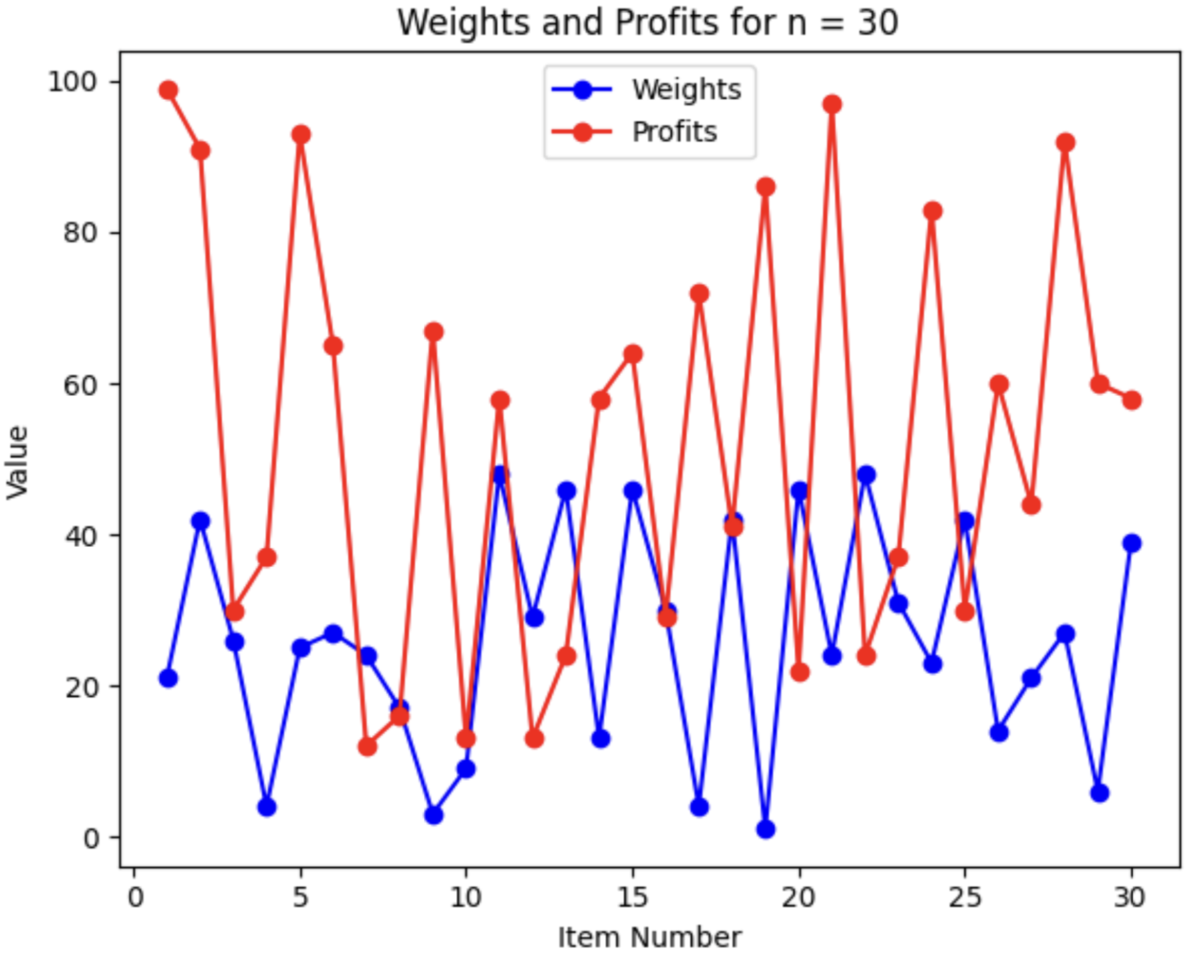

Figure 3: value for 30