

0/1 Knapsack Problem Example

Problem Statement

Given 4 items with the following weights and profits:

Item	Weight (w_i)	Profit (p_i)
1	2	1
2	3	2
3	4	5
4	5	6

Knapsack capacity $W = 8$. Determine the **maximum profit** and the items to include using **0/1 Knapsack with Dynamic Programming**.

Step 1: DP Table Definition

Let $dp[i][w]$ denote the maximum profit obtainable using the first i items and a knapsack of capacity w .

$$dp[i][w] = \begin{cases} 0 & \text{if } i = 0 \text{ or } w = 0 \\ \max(dp[i-1][w], p_i + dp[i-1][w - w_i]) & \text{if } w_i \leq w \\ dp[i-1][w] & \text{if } w_i > w \end{cases}$$

Step 2: DP Table Construction

i \ w	0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0	0
1	0	0	1	1	1	1	1	1	1
2	0	0	1	2	2	3	3	3	3
3	0	0	1	2	5	5	6	7	7
4	0	0	1	2	5	6	7	8	8

Step 3: Backtracking to Find Selected Items

Start from $dp[4][8] = 8$ and move backward:

- $dp[4][8] \neq dp[3][8] \rightarrow$ **Item 4 selected**, remaining capacity = $8 - 5 = 3$
- $dp[3][3] = dp[2][3] \rightarrow$ **Item 3 not selected**
- $dp[2][3] \neq dp[1][3] \rightarrow$ **Item 2 selected**, remaining capacity = $3 - 3 = 0$
- $dp[1][0] = 0 \rightarrow$ **Item 1 not selected**

Selected items: Item 2 and Item 4

Total profit: $2 + 6 = 8$

Total weight: $3 + 5 = 8$

—

Step 4: Summary

- Maximum Profit: 8
- Selected Items: 2 and 4
- DP table calculated maximum profit
- Backtracking used to identify which items to include

0/1 Knapsack Problem using Dynamic Programming

Problem Statement

Given the following items:

Item	Weight	Value
1	2	10
2	3	20
3	4	30
4	5	40

Knapsack capacity $W = 8$. Solve the problem using the **0/1 Knapsack Dynamic Programming** approach.

—

DP Formulation

Let $dp[i][w]$ denote the maximum value that can be obtained using the first i items with capacity w .

$$dp[i][w] = \begin{cases} 0, & i = 0 \text{ or } w = 0 \\ \max(dp[i-1][w], v_i + dp[i-1][w - w_i]), & w_i \leq w \\ dp[i-1][w], & w_i > w \end{cases}$$

Complete DP Table

i \ w	0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0	0
1	0	0	10	10	10	10	10	10	10
2	0	0	10	20	20	30	30	30	30
3	0	0	10	20	30	30	40	50	50
4	0	0	10	20	30	40	40	50	60

Backtracking to Find Selected Items

Starting from $dp[4][8] = 60$:

- $dp[4][8] \neq dp[3][8] \Rightarrow$ Item 4 selected Remaining capacity $= 8 - 5 = 3$
- $dp[3][3] = dp[2][3] \Rightarrow$ Item 3 not selected
- $dp[2][3] \neq dp[1][3] \Rightarrow$ Item 2 selected Remaining capacity $= 3 - 3 = 0$

Final Answer

- **Selected Items:** Item 2 and Item 4
- **Total Weight:** $3 + 5 = 8$
- **Maximum Value:** 60

0/1 Knapsack Problem using Dynamic Programming

Problem Statement

Given the following items:

Item	Weight	Profit
1	3	2
2	4	3
3	6	1
4	5	4

Knapsack capacity: $W = 8$.

DP Formulation

Let $dp[i][w]$ be the maximum profit obtained using the first i items with capacity w .

$$dp[i][w] = \begin{cases} 0, & i = 0 \text{ or } w = 0 \\ \max(dp[i-1][w], p_i + dp[i-1][w - w_i]), & w_i \leq w \\ dp[i-1][w], & w_i > w \end{cases}$$

Complete DP Table

i \ w	0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0	0
1	0	0	0	2	2	2	2	2	2
2	0	0	0	2	3	3	3	5	5
3	0	0	0	2	3	3	3	5	5
4	0	0	0	2	3	4	4	5	6

Backtracking to Find Selected Items

Start from $dp[4][8] = 6$:

- $dp[4][8] \neq dp[3][8] \Rightarrow$ Item 4 selected Remaining capacity $= 8 - 5 = 3$
- $dp[3][3] = dp[2][3] \Rightarrow$ Item 3 not selected
- $dp[2][3] = dp[1][3] \Rightarrow$ Item 2 not selected
- $dp[1][3] \neq dp[0][3] \Rightarrow$ Item 1 selected

Final Answer

- **Selected Items:** Item 1 and Item 4
- **Total Weight:** $3 + 5 = 8$
- **Maximum Profit:** 6