KNAPSACK PROBLEM

- The knapsack problem is given *n* items of known weights $w_1, \dots, w_n$ and values $v_1, \dots, v_n$ and a knapsack of capacity *W*, find the most valuable subset of the items that fit into the knapsack.
- This problem can also be solved using Dynamic Programming design technique.
- A recurrence relation has to be derived that expresses a solution to an instance of the knapsack problem in terms of solutions to its smaller subinstances.
- Divide all the subsets of the first *i* items that fit the knapsack of capacity *j* into two categories: those that do not include the ith item and those that do.
  - ✓ Among the subsets that do not include the *ith* item, the value of an optimal subset is, by definition, $V[i - 1, j]$.
  - ✓ Among the subsets that do include the ith item (hence, $j - w_i \geq 0$), an optimal subset is made up of this item and an optimal subset of the first $i - 1$ items that fit into the knapsack of capacity $j - w_i$. The value of such an optimal subset is $v_i + V[i - 1, j - w_i]$.
- Thus, the value of an optimal solution among all feasible subsets of the first *i* items is the maximum of these two values.
- This can be expressed as

$$V[i, j] = \begin{cases} \max\{V[i-1, j],\ v_i + V[i-1, j-w_i]\} & \text{if } j - w_i \geq 0 \\ V[i-1, j] & \text{if } j - w_i < 0. \end{cases}$$

with the initial condition

$$V[0, j] = 0 \text{ for } j \geq 0 \text{ and } V[i, 0] = 0 \text{ for } i \geq 0.$$

EXAMPLE:

Let us consider the given knapsack instance, with the sack capacity of W = 5

Item	**Weight**	**Value**
1	2	12
2	1	10
3	3	20
4	2	15

Solution

i	0	1	2	3	4	5
0	0	0	0	0	0	0
$w_1 = 2, v_1 = 12$ 1	0	0	12	12	12	12
$w_2 = 1, v_2 = 10$ 2	0	10	12	22	22	22
$w_3 = 3, v_3 = 20$ 3	0	10	12	22	30	32
$w_4 = 2, v_4 = 15$ 4	0	10	15	25	30	**37**

Fill first row and column entries by '0' based on the initial condition.

V[1,1] = V[0,1] = 0 {here, $j - w_i < 0$}

V[1,2] = max{V[0,2], v_1 + V[0,0]} = max{0, 12 + 0} = 12 {here, $j - w_i \geq 0$}

V[1,3] = max{V[0,3], v_1 + V[0,1]} = max{0, 12 + 0} = 12 {here, $j - w_i \geq 0$}

V[1,4] = max{V[0,4], v_1 + V[0,2]} = max{0, 12 + 0} = 12 {here, $j - w_i \geq 0$}

V[1,5] = max{V[0,5], v_1 + V[0,3]} = max{0, 12 + 0} = 12 {here, $j - w_i \geq 0$}

V[2,1] = max{V[1,1], v_2 + V[1,0]} = max{0, 10 + 0} = 10 {here, $j - w_i \geq 0$}

V[2,2] = max{V[1,2], v_2 + V[1,1]} = max{12, 10 + 0} = 12 {here, $j - w_i \geq 0$}

V[2,3] = max{V[1,3], v_2 + V[1,2]} = max{12, 10 + 12} = 22 {here, $j - w_i \geq 0$}

V[2,4] = max{V[1,4], v_2 + V[1,3]} = max{12, 10 + 12} = 22 {here, $j - w_i \geq 0$}

V[2,5] = max{V[1,5], v_2 + V[1,4]} = max{12, 10 + 12} = 22 {here, $j - w_i \geq 0$}

V[3,1] = V[2,1] = 10 {here, $j - w_i < 0$}

V[3,2] = V[2,2] = 12 {here, $j - w_i < 0$}

V[3,3] = max{V[2,3], v_3 + V[2,0]} = max{22, 20 + 0} = 22 {here, $j - w_i \geq 0$}

V[3,4] = max{V[2,4], v_3 + V[2,1]} = max{22, 20 + 10} = 30 {here, $j - w_i \geq 0$}

V[3,5] = max{V[2,5], v_3 + V[2,2]} = max{22, 20 + 12} = 32 {here, $j - w_i \geq 0$}

V[4,1] = V[3,1] = 10 {here, $j - w_i < 0$}

V[4,2] = max{V[3,2], v_4 + V[3,0]} = max{12, 15 + 0} = 15 {here, $j - w_i \geq 0$}

V[4,3] = max{V[3,3], v_4 + V[3,1]} = max{22, 15 + 10} = 25 {here, $j - w_i \geq 0$}

V[4,4] = max{V[3,4], v_4 + V[3,2]} = max{30, 15 + 12} = 30 {here, $j - w_i \geq 0$}

V[4,5] = max{V[3,5], v_4 + V[3,3]} = max{32, 15 + 22} = **37** {here, $j - w_i \geq 0$}

The maximum value of the sack is **37**

To find Solution Set

The composition of the optimal subset is obtained by tracing back the computations of the entry in the table.

Since V[4, 5] ≠ V[3, 5], item 4 was included in the optimal solution

The remaining 3 units of the knapsack capacity is represented by element V[3, 3]. Since V[3,3] = V[2, 3], item 3 is not a part of the optimal subset.

Since V[2, 3] ≠ V[l, 3], item 2 is a part of an optimal selection.

Similarly, since V[1, 2] ≠ V[0, 2], item 1 is the final part of the optimal solution.

Therefore, solution set **{I_1, I_2, I_4}.**