0/1 Knapsack Problem
Problem:

You are given a knapsack with a limited weight capacity and some items each of which have a weight and a value.

The problem is: "Which items to place in the knapsack such that the weight limit is not exceeded and the total value of the items is as large as possible?"
Knapsack Problem Variants

0/1 Knapsack Problem

- Items are indivisible i.e., you cannot break an item, you either take an item or not.
- Solved with a dynamic programming approach.

Fractional Knapsack Problem

- Items are divisible i.e., you can take any fraction of an item.
- Solved with a greedy approach.
Question: For the given set of items and Knapsack capacity = 5, find the optimal solution for the Knapsack 0/1 problem by using dynamic programming approach.

<table>
<thead>
<tr>
<th>Item</th>
<th>Weight</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>

OR

Find the optimal solution for the Knapsack 0/1 problem by using the dynamic programming approach. Consider:

\[ n = 4, \ W = 5, \ (w_1,w_2,w_3,w_4) = (2,3,4,5) \] and \[ (b_1, b_2, b_3, b_4) = (3,4,5,6) \]
OR

A thief enters a house for robbing it. He can carry a maximal weight of 5 Kg into his bag. There are four items in the house with the following weight and value. What items should thief take? He either takes or leaves the item.

<table>
<thead>
<tr>
<th>Item</th>
<th>Weight</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mirror</td>
<td>2 Kg</td>
<td>$3</td>
</tr>
<tr>
<td>Silver nugget</td>
<td>3 Kg</td>
<td>$4</td>
</tr>
<tr>
<td>Painting</td>
<td>4 Kg</td>
<td>$5</td>
</tr>
<tr>
<td>Vase</td>
<td>5 Kg</td>
<td>$6</td>
</tr>
</tbody>
</table>
How to proceed?

Consider we are given a knapsack of weight capacity \( W \) and \( n \) number of items with some weights.

Then, we proceed by drawing a Table \( 'T' \) with \((n+1)\) number of rows and \((W+1)\) number of columns.

Then, we fill all the boxes of 0th row and 0th column with zero as shown below:

\[
\begin{array}{cccccc}
0 & 0 & 0 & 0 & \ldots & 0 \\
0 & 0 & 0 & 0 & \ldots & 0 \\
0 & 0 & 0 & 0 & \ldots & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & 0 & \ldots & 0 \\
\end{array}
\]

T-table
Now, we start completing the table row wise from left to right using the following formula:

\[ T(i, j) = \max \{ T(i-1, j), \text{(value)}_i + T(i-1, j - \text{(weight)}_i) \} \]

**Meaning:**

\( T(i, j) \) = Maximum value of the selected items, if we are allowed to take items 1 to i and we have weight restrictions of j.
Solution:

Given:
- Knapsack Capacity \( (w) = 5 \)
- No. of items \( (n) = 4 \)

We start by drawing a table 'T' with \( (n+1) = 4+1 = 5 \) rows and \( (w+1) = (5+1) = 6 \) columns.

We then fill all the boxes of 0th row and 0th column with 0.
Now, we start completing the table row wise from left to right by using the following formula:

\[
T(i, j) = \max \{ T(i-1, j), (\text{value})_i + T(i-1, j-(\text{weight})_i) \}
\]

**Finding \( T(1,1) \):**

We have,

\[
\begin{align*}
  i &= 1 \\
  j &= 1 \\
  (\text{value})_i &= (\text{value})_1 = 3 \\
  (\text{weight})_i &= (\text{weight})_1 = 2
\end{align*}
\]

Substituting the values, we get:

\[
T(1,1) = \max \{ T(1-1, 1), 3 + T(1-1, 1-2) \}
\]

\[
= \max \{ T(0,1), 3 + T(0,-1) \}
\]

Ignore

\[
= T(0,1)
\]

\[
\therefore T(1,1) = 0
\]
Finding $T(1,2)$: we have,

\begin{align*}
i &= 1 \\
j &= 2
\end{align*}

\begin{align*}
(value)^*_i &= (value)_1 = 3 \\
(weight)^*_i &= (weight)_1 = 2
\end{align*}

Substituting the values, we get:

$$T(1,2) = \max \{ T(1-1,2), 3 + T(1-1, 2-2) \}$$

$$= \max \{ T(0,2), 3 + T(0,0) \}$$

$$= \max \{ 0, 3 + 0 \}$$

\[ \therefore T(1,2) = 3 \]

Finding $T(1,3)$: we have,

\begin{align*}
i &= 1 \\
j &= 3
\end{align*}

\begin{align*}
(value)^*_i &= (value)_1 = 3 \\
(weight)^*_i &= (weight)_1 = 2
\end{align*}
Substituting the values, we get:

\[
T(1, 3) = \max \{ T(1-1, 3) , 3 + T(1-1, 3-2) \} \\
= \max \{ T(0, 3) , 3 + T(0, 1) \} \\
= \max \{ 0 , 3 + 0 \} \\
= \max \{ 0, 3 \}
\]

\[\therefore T(1, 3) = 3\]

Finding \(T(1, 4)\):

We have:

\[
i = 1 \\
i = 4 \\
(value)_i = (value)_1 = 3 \\
(weight)_i = (weight)_1 = 2
\]

Substituting the values, we get:

\[
T(1, 4) = \max \{ T(1-1, 4) , 3 + T(1-1, 4-2) \} \\
= \max \{ T(0, 4) , 3 + T(0, 2) \} \\
= \max \{ 0, 3 + 0 \} \\
\]

\[\therefore T(1, 4) = 3\]
Finding $T(1,5)$: we have,

\[ t = 1 \]
\[ j = 5 \]
\[ (\text{value})_t = (\text{value})_1 = 3 \]
\[ (\text{weight})_t = (\text{weight})_1 = 2 \]

Substituting the values, we get:

\[
T(1,5) = \max \{ T(1-1, 5), 3 + T(1-1, 5-2) \} \]
\[
= \max \{ T(0,5), 3 + T(0,3) \} \]
\[
= \max \{ 0, 3+0 \} \]
\[
= \max \{ 0,3 \} \]

\[ \therefore T(1,5) = 3 \]
Finding $T(2,1)$:

We have,

$i = 2$

$j = 1$

$(\text{value})_i = (\text{value})_2 = 4$

$(\text{weight})_i = (\text{weight})_2 = 3$

Substituting the values, we get:

$$T(2,1) = \max \{ T(2-1,1), 4 + T(2-1,1-3) \}$$

$$= \max \{ T(1,1), 4 + T(1,-2) \}$$

$\downarrow$

Ignore

$$= T(1,1)$$

So $T(2,1) = 0$.
Finding $T(2,2)$: we have, \[ i = 2 \]
\[ j = 2 \]
\[ (\text{value})_i = (\text{value})_2 = 4 \]
\[ (\text{weight})_i = (\text{weight})_2 = 3 \]

Substituting the values, we get:
\[
T(2,2) = \max \{ T(2-1,2), 4 + T(2-1,2-3) \}
\]
\[
= \max \{ T(1,2), 4 + T(1,-1) \}
\]
\[
\downarrow \quad \text{Ignore}
\]
\[
= T(1,2)
\]

Thus, $T(2,2) = 3$.
Finding $T(2,3)$:

We have,

\[ i = 2, \quad j = 3 \]

\[ (\text{value})_i = (\text{value})_2 = 4 \]

\[ (\text{weight})_i = (\text{weight})_2 = 3 \]

Substituting the values, we get:

\[ T(2,3) = \max \{ T(2-1,3), 4 + T(2-1,3-3) \} \]

\[ = \max \{ T(1,3), 4 + T(1,0) \} \]

\[ = \max \{ 3, 4+0 \} \]

\[ = \max \{ 3, 4 \} \]

\[ \therefore T(2,3) = 4 \]
Finding $T(2, 4)$:

We have,

\begin{align*}
&i = 2 \\
&j = 4 \\
&\text{(value)}_i = \text{(value)}_j = 4 \\
&\text{(weight)}_i = \text{(weight)}_j = 3
\end{align*}

Substituting the values, we get:

\[
T(2, 4) = \max \{ T(2-1, 4), 4 + T(2-1, 4-3) \} \\
= \max \{ T(1, 4), 4 + T(1, 1) \} \\
= \max \{ 3, 4 + 0 \} \\
= \max \{ 3, 4 \}
\]

\[
\therefore T(2, 4) = 4
\]
Finding \( T(2,5) \): We have,

\[
\begin{align*}
  i &= 2 \\
  j &= 5 \\
  \text{value}_1 &= \text{value}_2 = 4 \\
  \text{weight}_1 &= \text{weight}_2 = 3
\end{align*}
\]

Substituting the values, we get-

\[
T(2,5) = \max \{ T(2-1,5), 4 + T(2-1, 5-3) \}
\]

\[
= \max \{ T(1,5), 4 + T(1,3) \}
\]

\[
= \max \{ 3, 4+3 \}
\]

\[
= \max \{ 3, 7 \}
\]

\[
\therefore T(2,5) = 7
\]
Finally, after computing all the values and filling them in the table, we get:

```
  0  1  2  3  4  5
0  0  0  0  0  0  0
1  0  0  3  3  3  3
2  0  0  3  4  4  7
3  0  0  3  4  5  7
4  0  0  3  4  5  7
```

This entry represents the maximum possible value which can be put in a Knapsack.

```
:: Maximum possible value which can be put in a Knapsack = 7
```
Finding the items to be put in a Knapsack to get maximum possible value 7:

We consider the last column of the table and start scanning the entries from bottom to top. If we encounter an entry whose value is not same as the value which is stored in the entry immediately above it, then we will mark the label of row of that entry.

Following this, we mark the rows labelled "1" and "2".

So, the items to be put in a Knapsack to obtain the maximum value 7 are:

Item 1 and Item 2
Analysis:

This dynamic 0-1 knapsack algorithm takes $\Theta(nw)$ times broken up as follows:

- $\Theta(nw)$ time to fill the table having $(n+1)(w+1)$ entries, each requiring $\Theta(1)$ time to compute.
- $\Theta(n)$ time to trace the solution, because the tracing process starts in row $n$ of the table and moves up 1 row at each step.