

**Example:**

Apply dynamic programming algorithm to the instance of the knapsack problem shown in the table. Assume that the Knapsack capacity is M=3.

|  |  |  |
| --- | --- | --- |
| Item | Weight | Value |
| 1 | 1 | 1 |
| 2 | 2 | 6 |
| 3 | 4 | 4 |

**Solution**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | j | | | |
|  |  | 0 | 1 | 2 | 3 |
| i | 0 |  |  |  |  |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |

**Step 1**

V[0,j] = 0 for all j>=0

V[i,0] = 0 for all i>=0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | j | | | |
|  |  | 0 | 1 | 2 | 3 |
| i | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 |  |  |  |
| 2 | 0 |  |  |  |
| 3 | 0 |  |  |  |

**Step 2**

1ST Item Weight, wi = 1 Value vi=1

V[1,1] i=1 j=1 wi=1 vi=1 wi = j

V[i,j] = max { V[i-1,j], vi + V[i-1,j-wi]}

V[1,1] = max { V[i-1,j], vi + V[i-1,j-wi]}

= max{ V[0,1], 1 + V[0,0]}

= max { 0 , 1+0 }

= 1

V[1,2] i=1 j=2 wi=1 vi=1 wi < j

V[i,j] = max { V[i-1,j], vi + V[i-1,j-wi]}

V[1,1] = max { V[i-1,j], vi + V[i-1,j-wi]}

= max{ V[0,2], 1 + V[0,1]}

= max { 0 , 1+0 }

= 1

V[1,3] i=1 j=3 wi=1 vi=1 wi < j

V[i,j] = max { V[i-1,j], vi + V[i-1,j-wi]}

V[1,1] = max { V[i-1,j], vi + V[i-1,j-wi]}

= max{ V[0,3], 1 + V[0,2]}

= max { 0 , 1+0 }

= 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | j | | | |
|  |  | 0 | 1 | 2 | 3 |
| i | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 2 | 0 |  |  |  |
| 3 | 0 |  |  |  |

**Step 3**

2ST Item Weight, wi = 2 Value vi=6

V[2,1] i=2 j=1 wi=2 vi=6 wi > j

V[i,j] = V[i-1,j]

V[2,1] = V [i-1, j]

= V[1,1]

= 1

V[2,2] i=2 j=2 wi=2 vi=6 wi = j

V[i,j] = max { V[i-1,j], vi + V[i-1,j-wi]}

V[2,2] = max { V[i-1,j], vi + V[i-1,j-wi]}

= max { V[1,2] , 6 + V[1,0]}

= max { 1, 6+0 }

= 6

V[2,3] i=2 j=3 wi=2 vi=6 wi = j

V[i,j] = max { V[i-1,j], vi + V[i-1,j-wi]}

V[2,3] = max { V[i-1,j], vi + V[i-1,j-wi]}

= max { V[1,3] , 6 + V[1,3]}

= max { 1, 6+1 }

= 7

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | j | | | |
|  |  | 0 | 1 | 2 | 3 |
| i | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 2 | 0 | 1 | 6 | 7 |
| 3 | 0 |  |  |  |

**Step 4**

3rd Item Weight, wi = 4 Value vi=4

V[3,1] i=3 j=1 wi=4 vi=4 wi > j

V[i,j] = V[i-1,j]

V[3,1] = V [i-1, j]

= V[2,1]

= 1

V[3,2] i=3 j=2 wi=4 vi=4 wi > j

V[i,j] = V[i-1,j]

V[3,2] = V [i-1, j]

= V[2,2]

= 6

V[3,3] i=3 j=3 wi=4 vi=4 wi > j

V[i,j] = V[i-1,j]

V[3,3] = V [i-1, j]

= V[2,3]

= 7

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | j | | | |
|  |  | 0 | 1 | 2 | 3 |
| i | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 2 | 0 | 1 | 6 | 7 |
| 3 | 0 | 1 | 6 | 7 |

Pick the solution

Item 3 – Rejected

Item 2, 3 - Accepted