Shri Vile Parle Kelavani Mandal's

Institute Of Technology, Dhule

Department of Information Technology

Subject : Design and Analysis of Algorithm lab

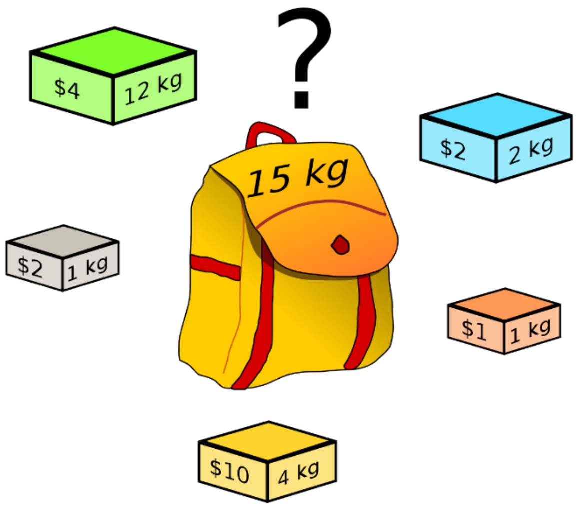
Assignment Title : 0/1 Knapsack Problem

Student Name	Roll No	PRN
Vansh Sambhaji Gosavi	13	2254491246013

Title: **0/1 Knapsack**

❖What is Knapsack ?

Knapsack is like a container Or a bag. Suppose we have some items which are having some weights and profit. We have to put some items in the knapsack in such a way that total value produces a maximum profit.



For example: the weight of the container is 15 Kg. We have to select the items in such a way that sum of the weight of all items is smaller or equal to 15 and the profit should be maximum.

There are two types of Knapsack:

- Fractional Knapsack
- 0/1 Knapsack

We are going to see 0/1 Knapsack ….

❖What is 0/1 Knapsack ?

Suppose a thief wants to rob a store. He is carrying a knapsack(bag) of capacity some 'W'. The store has 'n' items. Its weight is given by the 'wt' array and its value by the 'val' array.

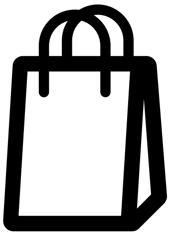
He can either include an item in the bag or exclude it but can't have fraction of item. We have to find the maximum number of items the thief can steal.

We use the approach of Dynamic Programming for 0/1 Knapsack Problem.

- **Dynamic Approach :** It is a programming technique where an algorithmic problem is first broken down into sub-problems, the results are saved, and the sub-problems are optimized to solve the overall problem

Example:

item	**weight**	**profit**
1	2	3
2	3	4
3	4	5
4	5	6



Capacity of bag = W = 5kg

- number of items = n = 4
- weight of items = w
- profit = p

Sort the weights of the items if required, and write the profit aligned to it.

Sorted weights and profit aligned to it:

weight	2	3	4	5
profit	3	4	5	6

Step 1: Draw a table ($T_{i \times w}$)

Draw a table 'T' with :

- Number of rows (n+1) = 4+1 = 5.
- Number of columns (w+1) = 5+1 = 6.
- Fill the 0^{th} row(i=0) and 0^{th} column(W=0) with 0.

W

i

	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**

Step 2: Fill the first item in the bag.

Formula to be used:

$$T[i,W] = \max(T[i-1,W],\ T[i-1,\ W-w[i]] + p[i])$$

Solution:

1.) i=1, p=3

- i=1, W=2

T[1,2] = max(T[0,2] , T[0,0] + 3)

T[1,2] = max(0 , 3)

T[1,2] = 3

- i=1, W=3

T[1,2] = max(T[0,3] , T[0,1] + 3)

T[1,2] = max(0 , 3)

T[1,2] = 3

- i=1, W=4

T[1,2] = max(T[0,4] , T[0,2] + 3)

T[1,2] = max(0 , 3)

T[1,2] = 3

- i=1, W=5

T[1,2] = max(T[0,5] , T[0,3] + 3)

T[1,2] = max(0 , 3)

T[1,2] = 3

2.) i=2 , p=4

- i=2, W=3

T[2,3] = max(T[1,3] , T[1,0] + 4)

T[2,3] = max(3,4)

T[2,3] = 4

- i=2, W=4

T[2,4] = max(T[1,4] , T[1,1] + 4)

T[2,4] = max(3 , 4)

T[2,4] = 4

- i=2, W=5

T[2,5] = max (T[1,5], T[1,2]+4 }

T[2,5] = max (3 , 4+3)

T[2,5] = 7

3.) i=3 , p=7

- i=3, W=4

T[3,4] = max(T[2,4] , T[2,4-4]+5)

T[3,4] = max(4 , 5)

T[3,4] = 5

- i=3 , W=5

T[3,5] = max(T[2,5] , T[2, 5-4]+5)

T[3,5] = max(7 , 6)

T[3,5] = 7

3.) i=3 , p=7

- i=4 , W=5

T[4,5]= max(T[3,5] , T[3,5-5]+6)

T[4,5]= max(7, 6)

T[4,5]=7

The last entry represents the maximum possible value that can be put into the knapsack .

So, maximum possible value that can be put into the knapsack = 7.