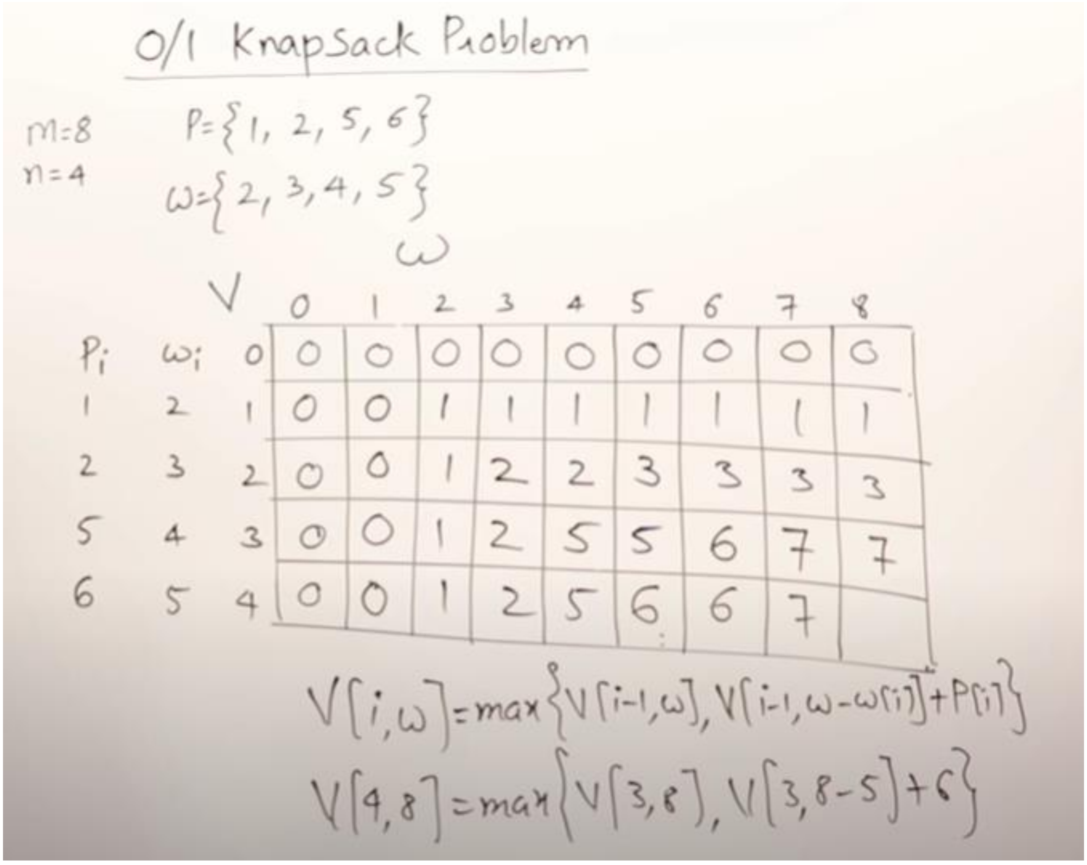
Tabulation Method



O/1 Knapsack Problem

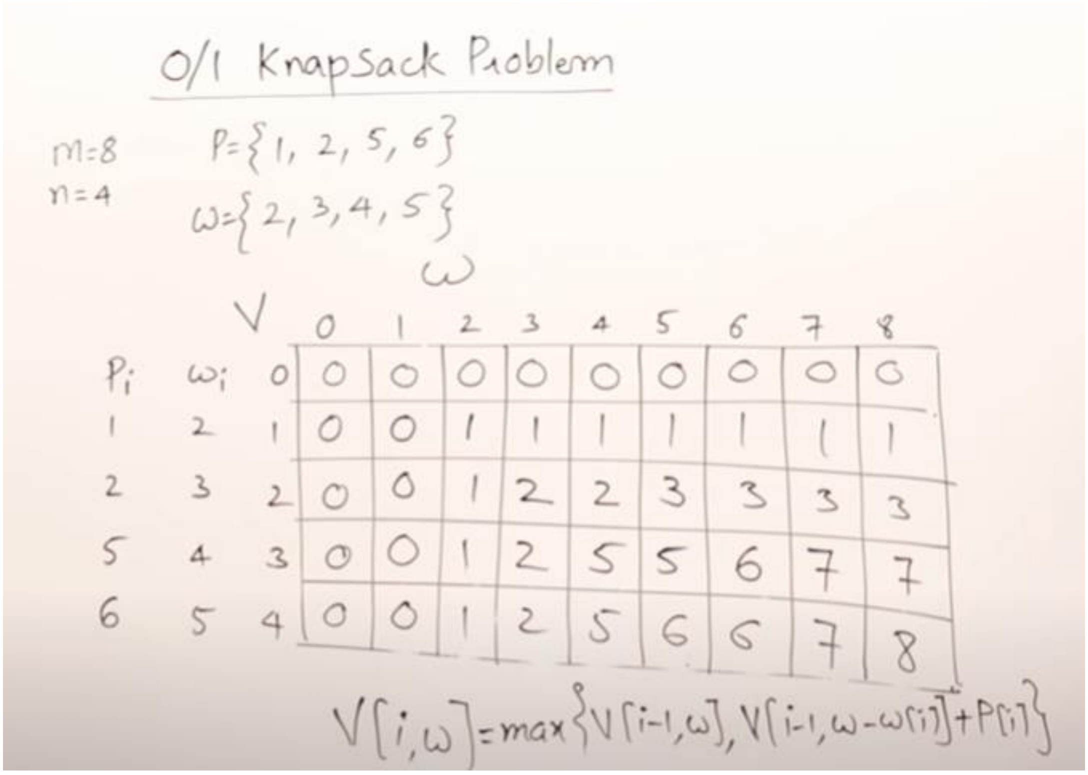
$m=8$ $\quad P=\{1, 2, 5, 6\}$

$n=4$ $\quad \omega=\{2, 3, 4, 5\}$

P_i	ω_i	V	0	1	2	3	4	5	6	7	8
		0	0	0	0	0	0	0	0	0	0
1	2	1	0	0	1	1	1	1	1	1	1
2	3	2	0	0	1	2	2	3	3	3	3
5	4	3	0	0	1	2	5	5	6	7	7
6	5	4	0	0	1	2	5	6	6	7	

$$V[i,\omega]=\max\{V[i-1,\omega], V[i-1,\omega-\omega[i]]+P[i]\}$$

$$V[4,8]=\max\{V[3,8], V[3,8-5]+6\}$$



O/1 Knapsack Problem

$m=8$ $\quad P=\{1, 2, 5, 6\}$

$n=4$ $\quad \omega=\{2, 3, 4, 5\}$

P_i	ω_i	V	0	1	2	3	4	5	6	7	8
		0	0	0	0	0	0	0	0	0	0
1	2	1	0	0	1	1	1	1	1	1	1
2	3	2	0	0	1	2	2	3	3	3	3
5	4	3	0	0	1	2	5	5	6	7	7
6	5	4	0	0	1	2	5	6	6	7	8

$$V[i,\omega]=\max\{V[i-1,\omega], V[i-1,\omega-\omega[i]]+P[i]\}$$

0/1 Knapsack Problem

$m=8$ $\quad P=\{1, 2, 5, 6\}$

$n=4$ $\quad W=\{2, 3, 4, 5\}$

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

$$\begin{Bmatrix} x_1 & x_2 & x_3 & x_4 \\ 0 & 1 & 0 & 1 \end{Bmatrix} \qquad \begin{aligned} 8-6&=2 \\ 2-2&=0 \end{aligned}$$

Set Method

0/1 Knapsack Problem

$m=8$ $\quad P=\{1, 2, 5, 6\}$

$n=4$ $\quad W=\{2, 3, 4, 5\}$

$\{0, 1, 0, 1\}$

$S^0=\{(0,0)\}$

$S^0_1=\{(1,2)\}$

$S^1=\{(0,0),(1,2)\}$

$S^1_1=\{(2,3),(3,5)\}$

$S^2=\{(0,0)(1,2)(2,3)(3,5)\}$

$S^2_1=\{(5,4),(6,6),(7,7),\cancel{(8,9)}\}$

$S^3=\{(0,0)(1,2)(2,3)\cancel{(3,5)}(5,4)(6,6)(7,7)\}$

$S^3_1=\{(6,5)(7,7)(8,8)\cancel{(11,9)}\cancel{(12,11)}\cancel{(13,12)}\}$

$S^4=\{(0,0)(1,2)(2,3)(5,4)\cancel{(6,6)}(6,5)(7,7)(8,8)\}$

① $(8,8)\in S^4$

but $(8,8)\notin S^3$ $\therefore x_4=1$

$(8-6,8-5)=(2,3)$

② $(2,3)\in S^3$ $\therefore x_3=0$

and $(2,3)\in S^2$

③ $(2,3)\in S^2$ $\therefore x_2=1$

but $(2,3)\notin S^1$

④ $(2-2,3-3)=(0,0)$

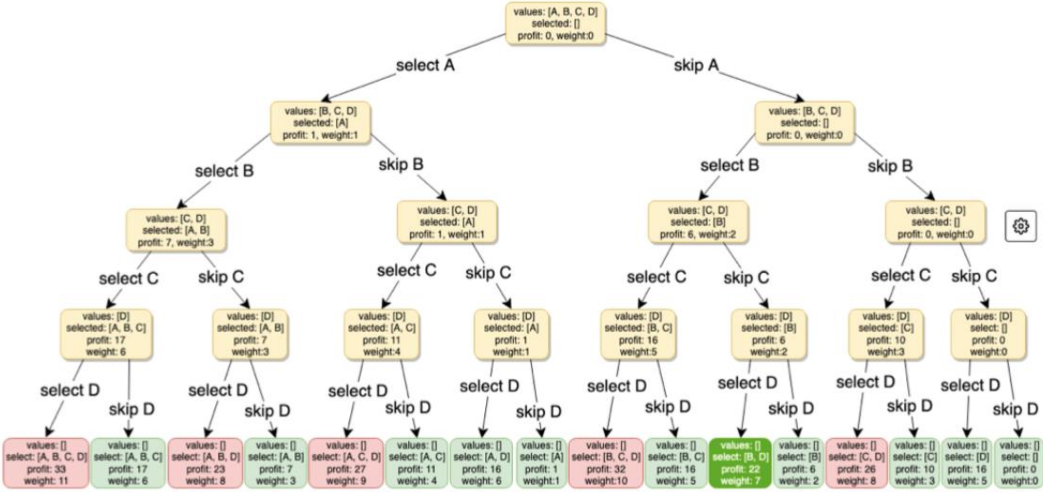
$(0,0)\in S^1$ and $(0,0)\in S^0$ $\therefore x_1=0$

Brute-Force : Recursive Solution

- Try out all possible combinations of items which has cumulative weight less than given capacity and then pick one with highest profit.
- For every item take 2 possiblities first with including this item (if weight constraint is fine) and second with excluding this item.
- Finally we can get the set with max profit following the capacity constriant.

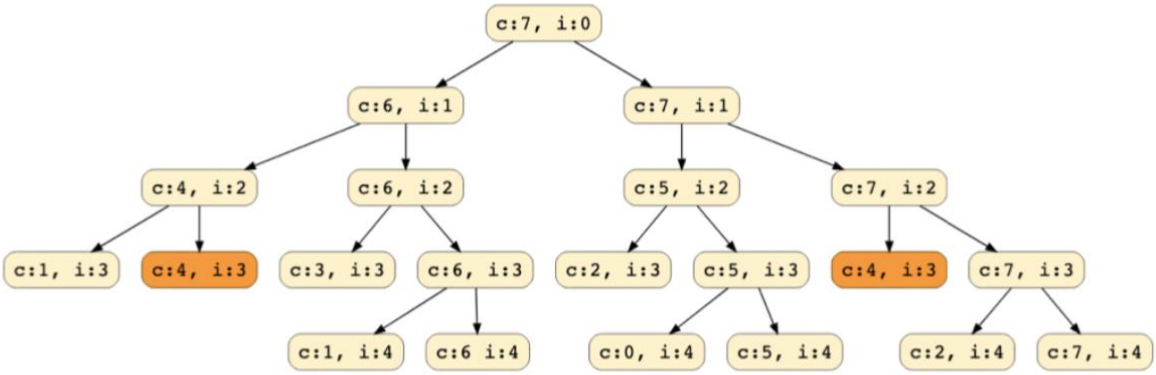
items	A	B	C	D
profit	1	6	10	16
weight	1	2	3	5

Capacity: 7



Identifying the problem as DP:

- Need to draw the recursive calls to see if there are any overlapping subproblems.
- In ***each recursive call, profits and weights array remain constant*** and only ***capacity and item index changes***.
- Drawing recursive calls with denoting index as ***i*** and capacity as ***c***.



- Here we see that overlapping sublproblems as `c:4, i:3` is repeating and hence can be solved using memoization.

Algorithm 1: Dynamic Programming Algorithm for 0-1 Knapsack Problem

Data: $W, v_1, v_2, v_3, ..., v_n$, $w_1, w_2, w_3, ..., w_n$

Result: $M[n, W]$

$M[0, w] \leftarrow 0,\ \forall w\ 0\ to\ W$

$M[i, 0] \leftarrow 0,\ \forall i\ 0\ to\ n$

```
for i ← 1 to n do
    for w ← 1 to W do
        if w_i ≤ w then
            M[i, w] = max(M[i − 1, w − w_i] + v_i, M[i − 1, w])
        else
            M[i, w] = M[i − 1, w]
        end
    end
end
return M[n, W]
```

Q. : What is the Time Complexity of 0/1 Knapsack Problem?

Ans: The time complexity for the 0/1 Knapsack problem solved using DP is O(N*W) where N denotes the number of items available and W denotes the capacity of the knapsack.