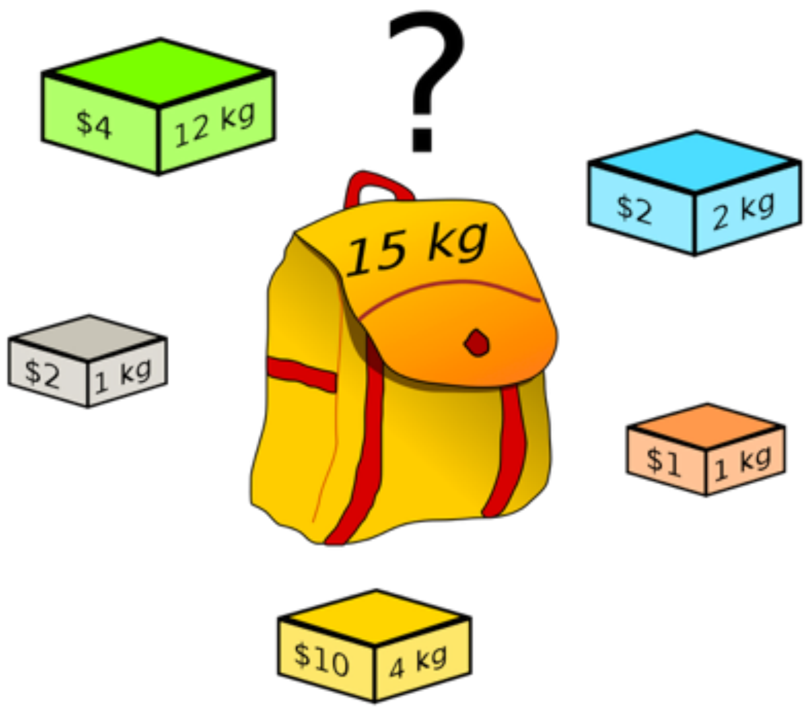
Knapsack

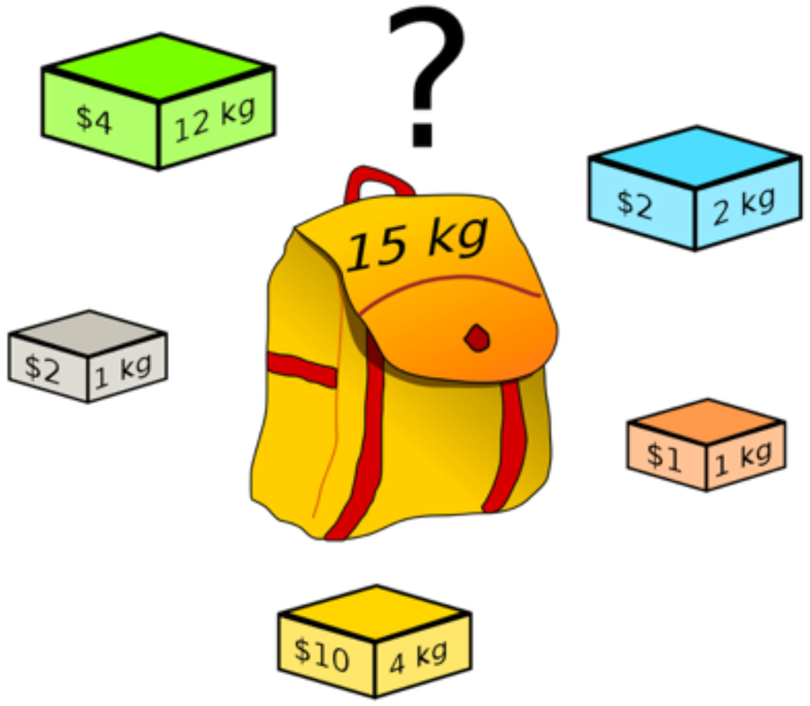
Shusen Wang

Knapsack Problem



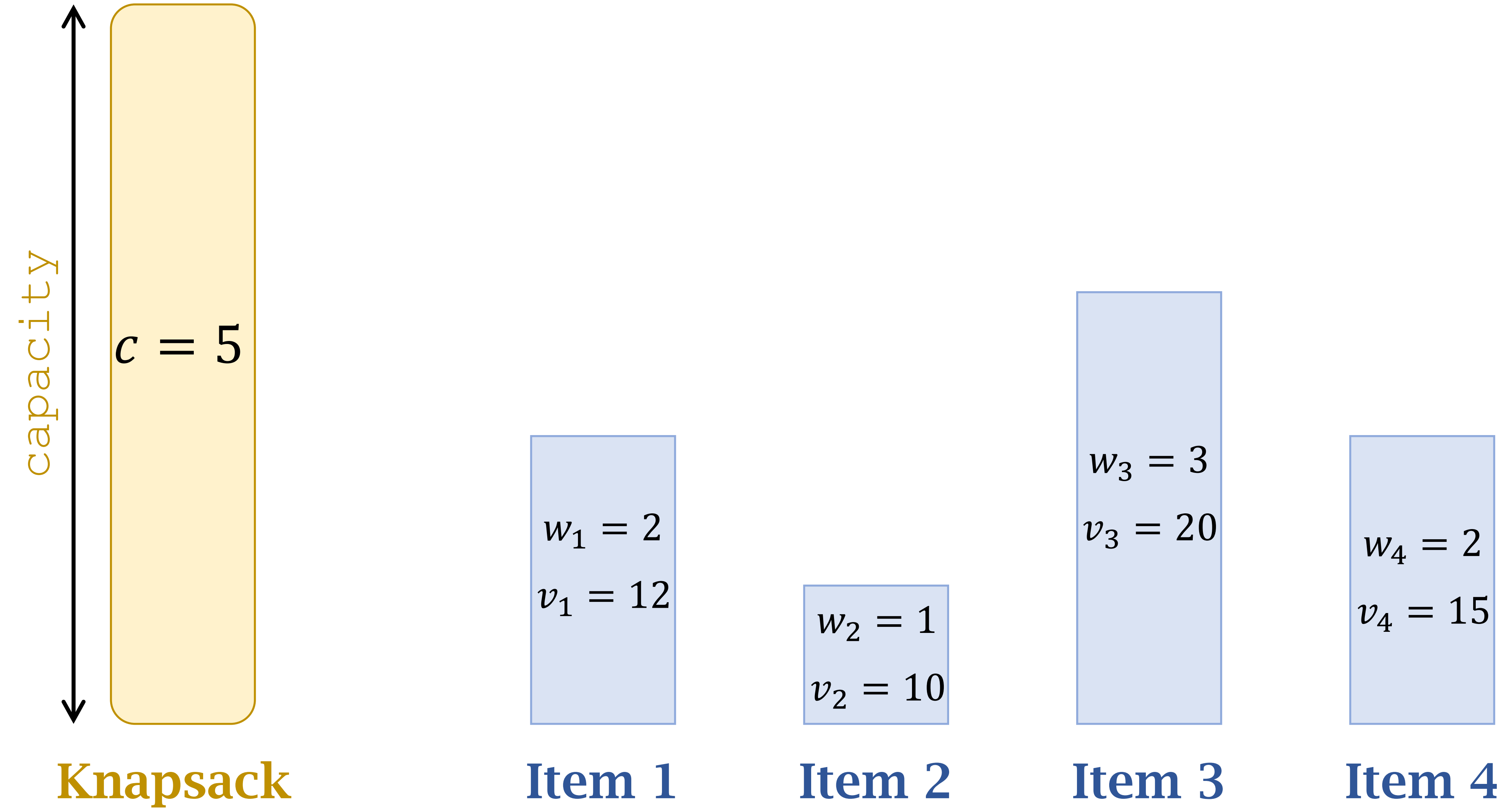
- **Inputs:**
  - A knapsack with capacity c.
  - A set of m items; the i-th item has weight w_i and value v_i.

Knapsack Problem



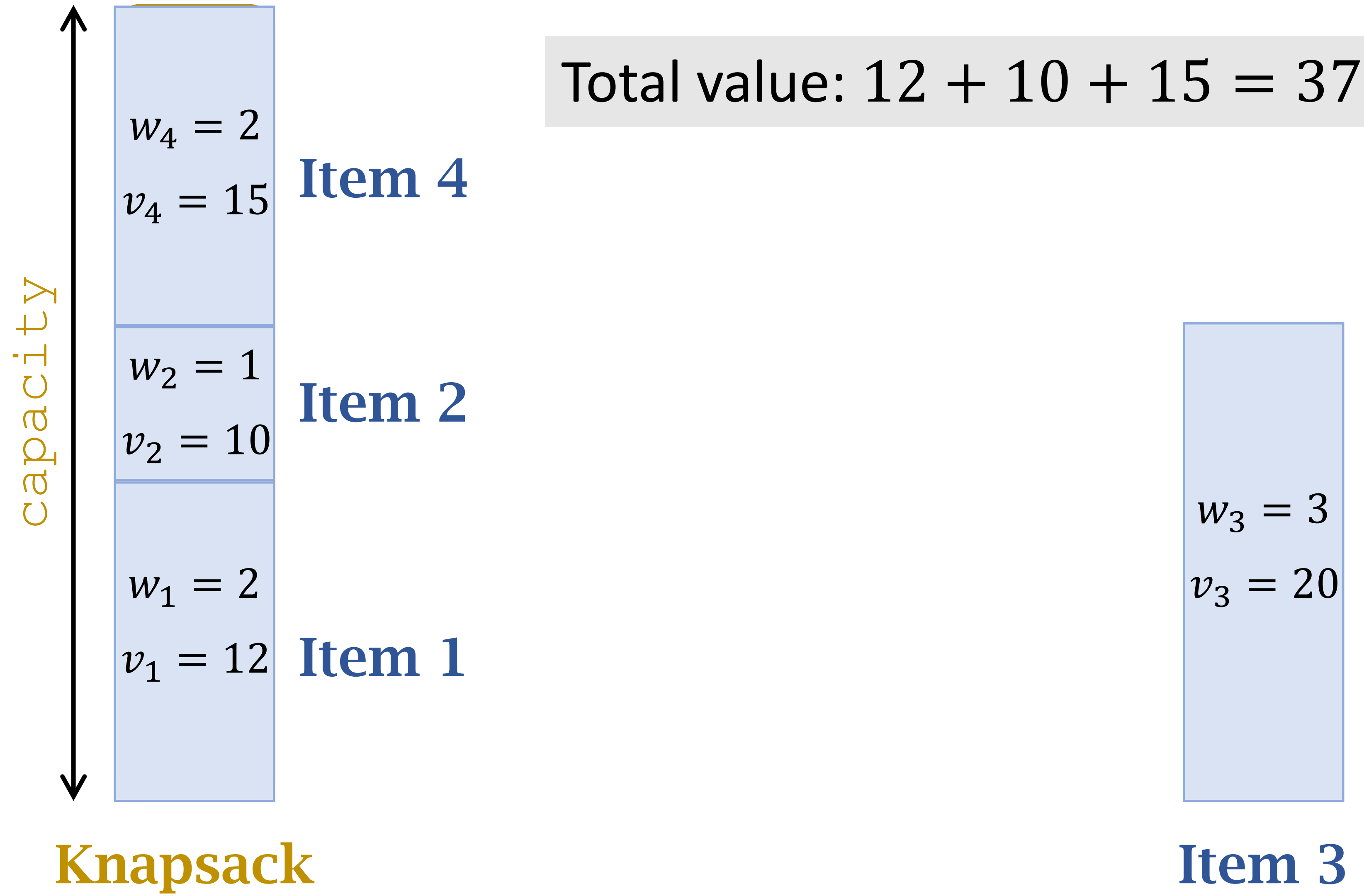
- **Inputs:**
  - A knapsack with capacity c.
  - A set of m items; the i-th item has weight w_i and value v_i.
- **Objective:** Find the most valuable subset of items that fits in the knapsack.
- **Constraint:** Sum of weights does not exceed c.

Inputs

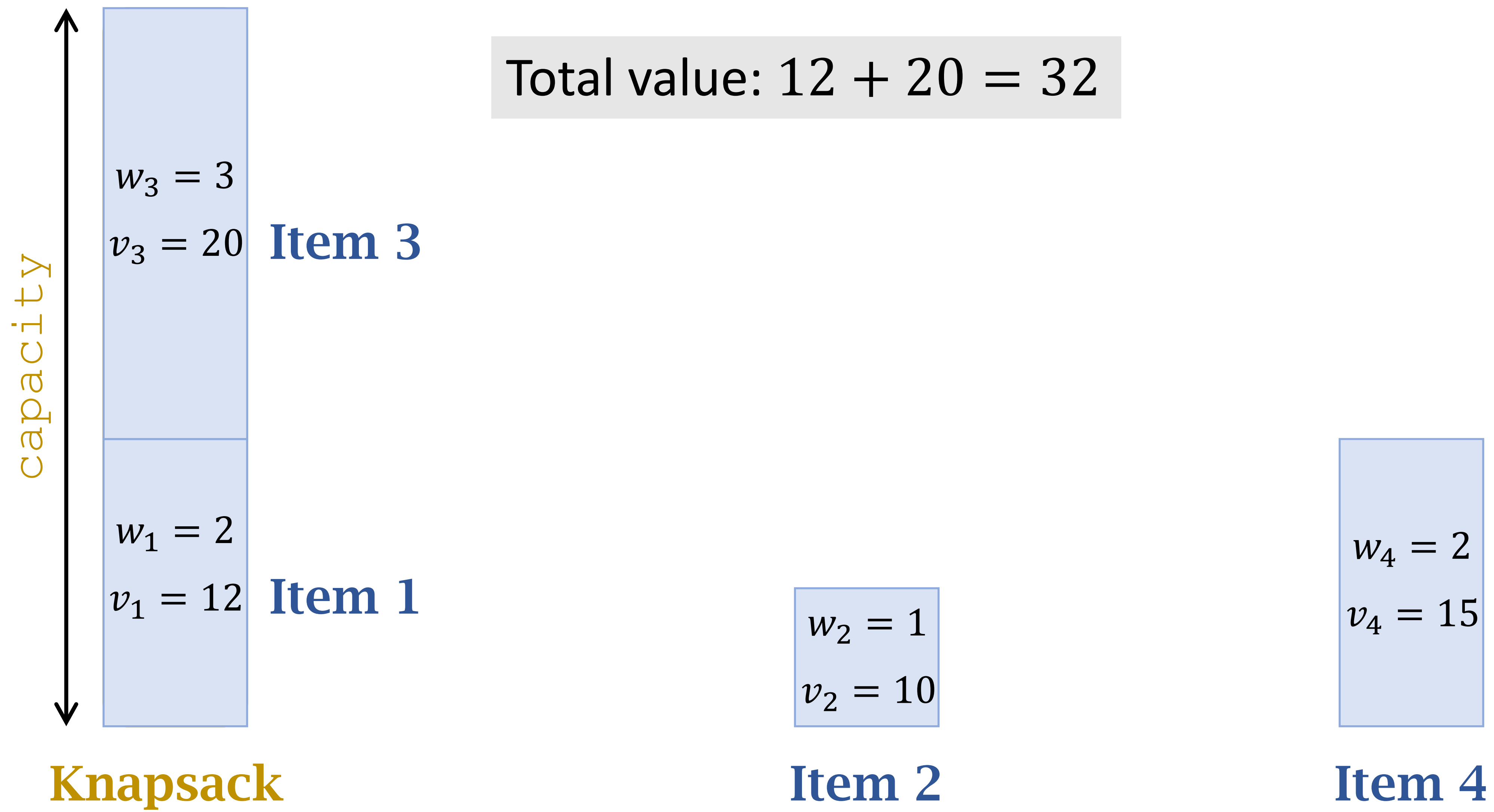

capacity
$c = 5$
$w_1 = 2$
$v_1 = 12$
$w_2 = 1$
$v_2 = 10$
$w_3 = 3$
$v_3 = 20$
$w_4 = 2$
$v_4 = 15$
Knapsack
Item 1
Item 2
Item 3
Item 4

Example 1

Total value: $12 + 10 + 15 = 37$

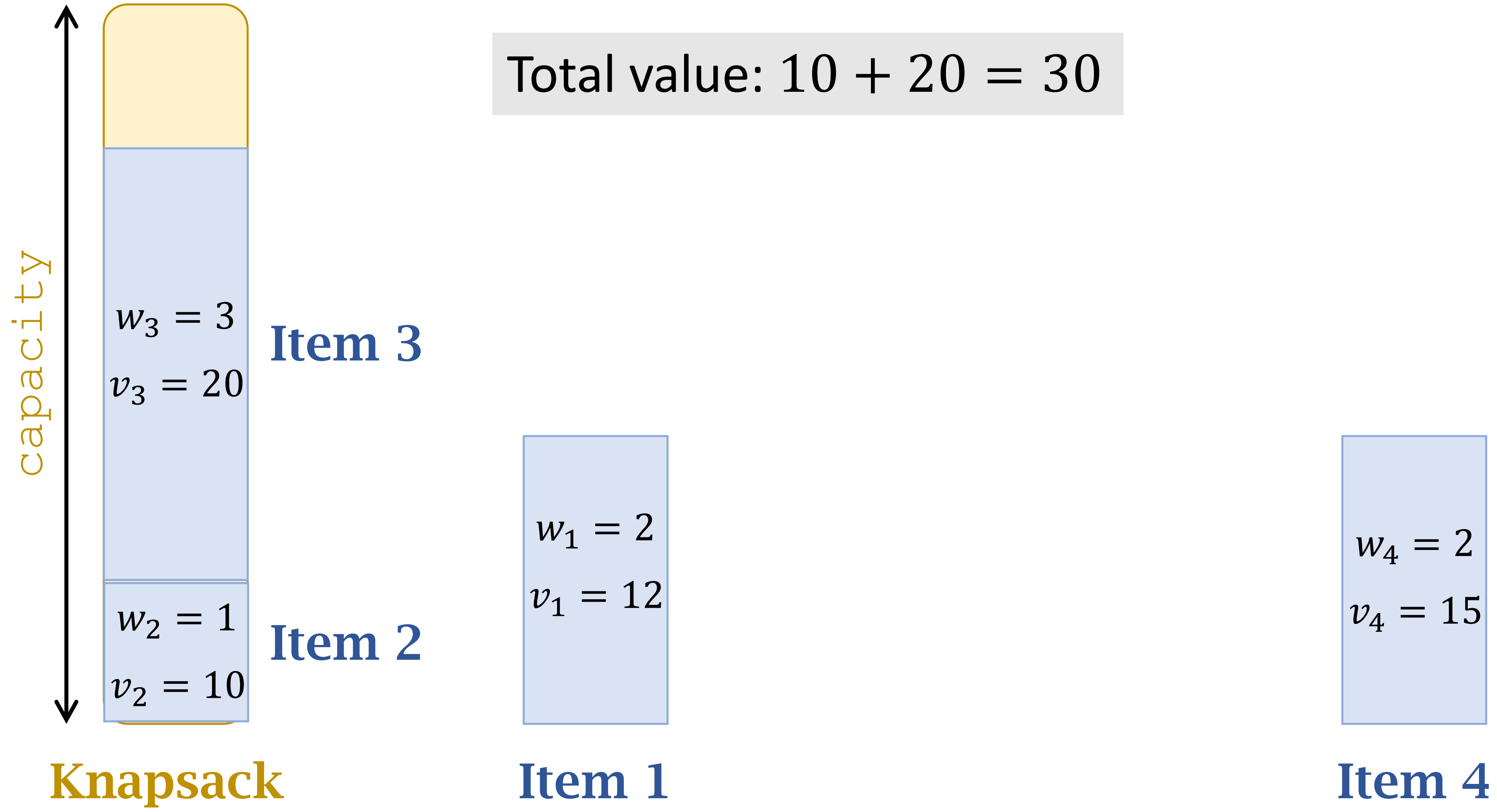


Example 2


Total value: $12 + 20 = 32$
capacity
$w_3 = 3$
$v_3 = 20$
Item 3
$w_1 = 2$
$v_1 = 12$
Item 1
$w_2 = 1$
$v_2 = 10$
$w_4 = 2$
$v_4 = 15$
Knapsack
Item 2
Item 4

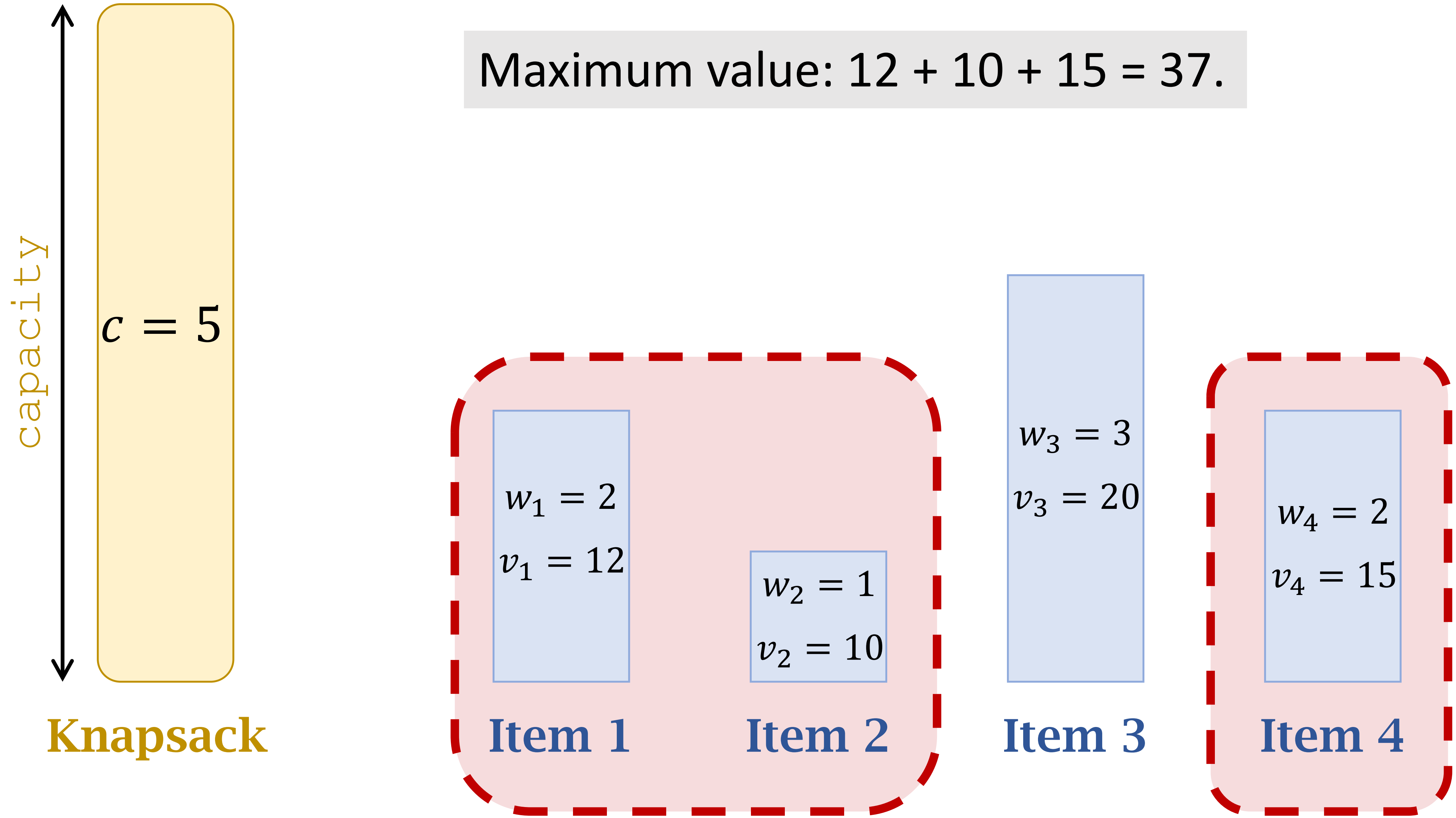
Example 3

Total value: $10 + 20 = 30$



Maximum Value

Maximum value: 12 + 10 + 15 = 37.



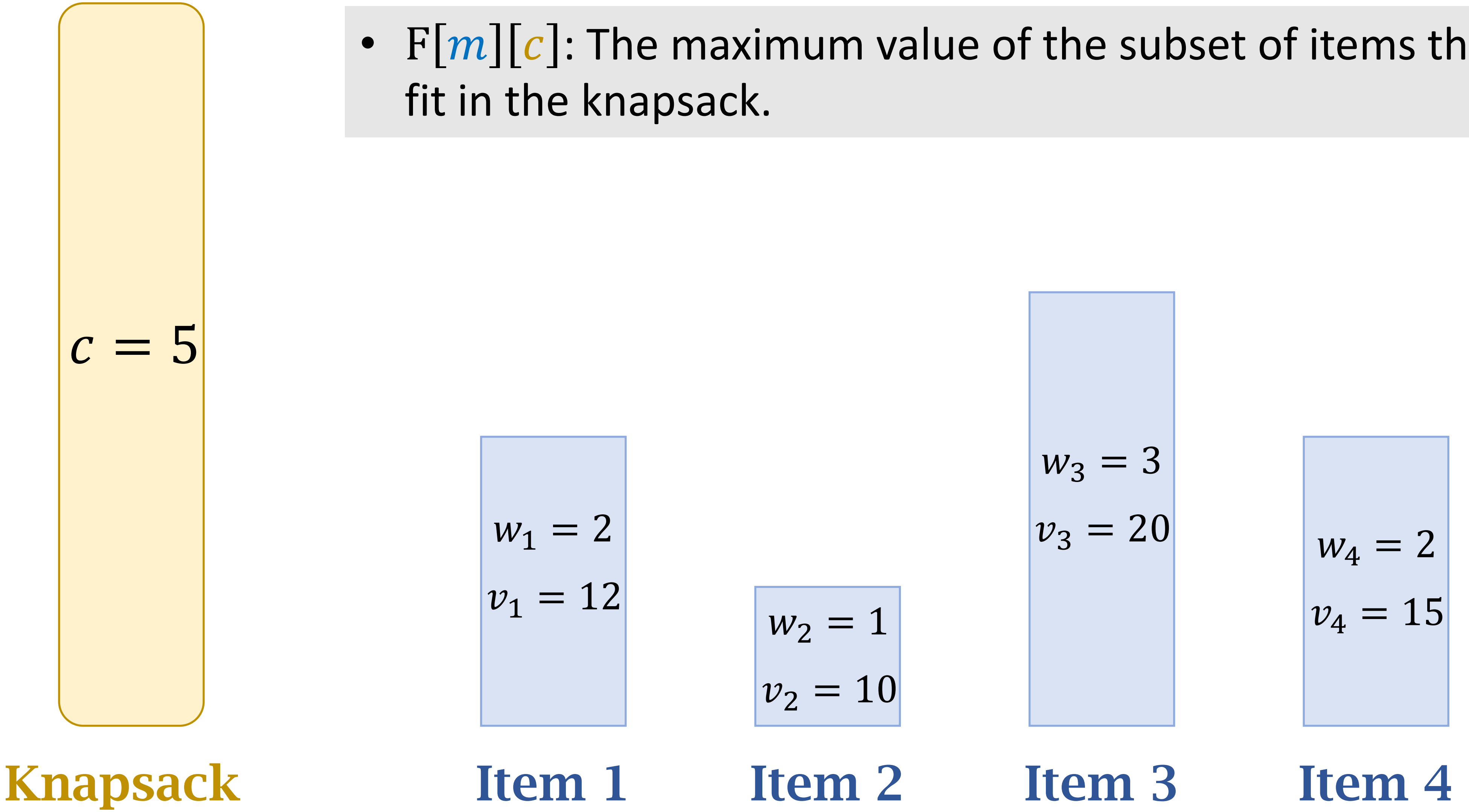
Maximum Value

- There are m items which have weights and values.
- The capacity of knapsack is c.
- Select a subset of items such that the total weight does not exceed c.
- Let $F[m][c]$ be the total value of the most valuable subset.

Optimal Substructure

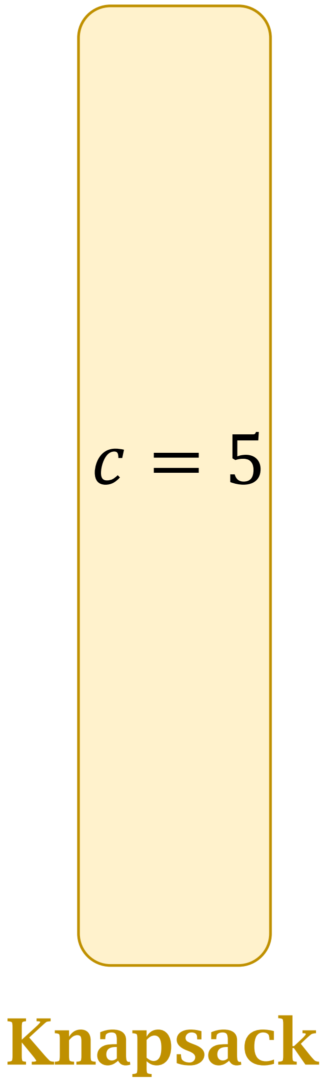
Original Problem

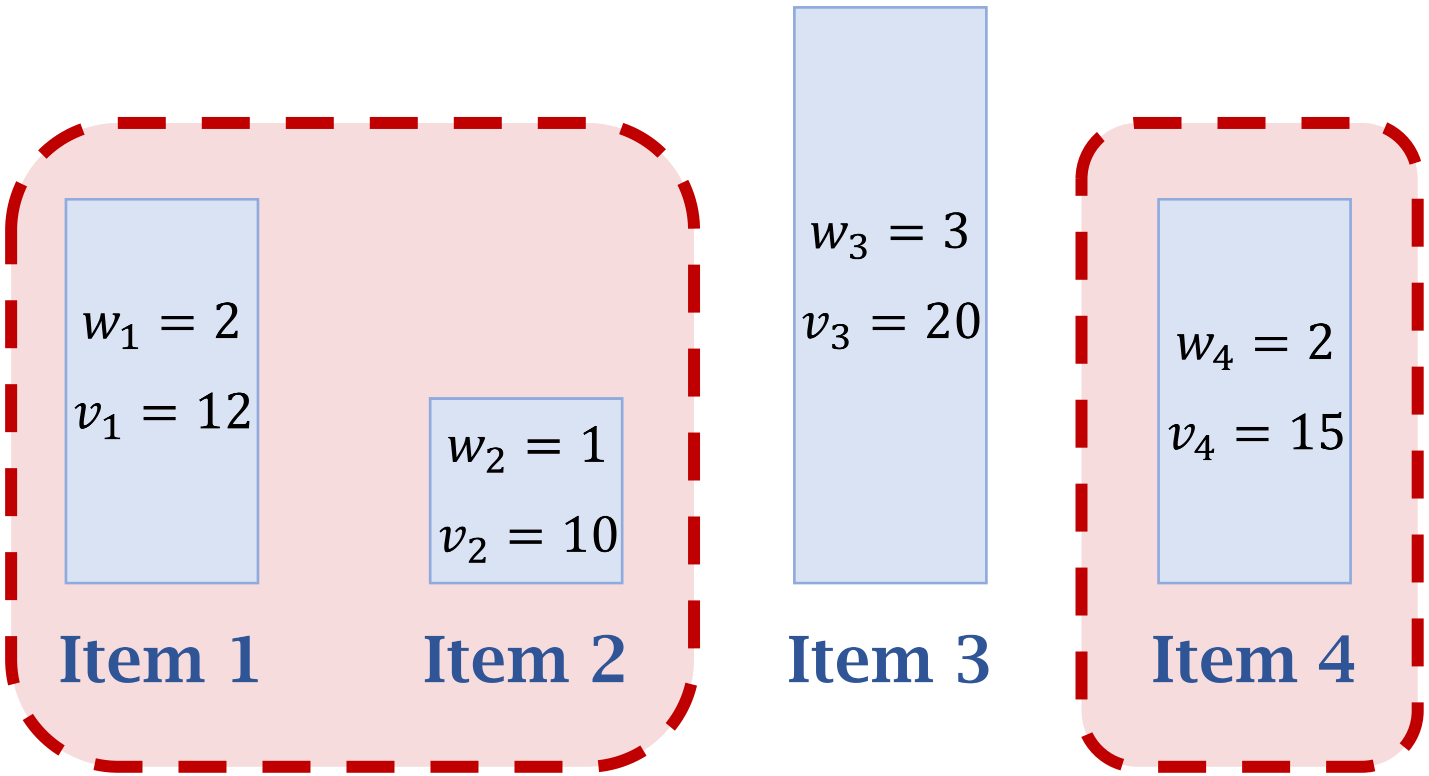
- F[m][c]: The maximum value of the subset of items that fit in the knapsack.



Original Problem

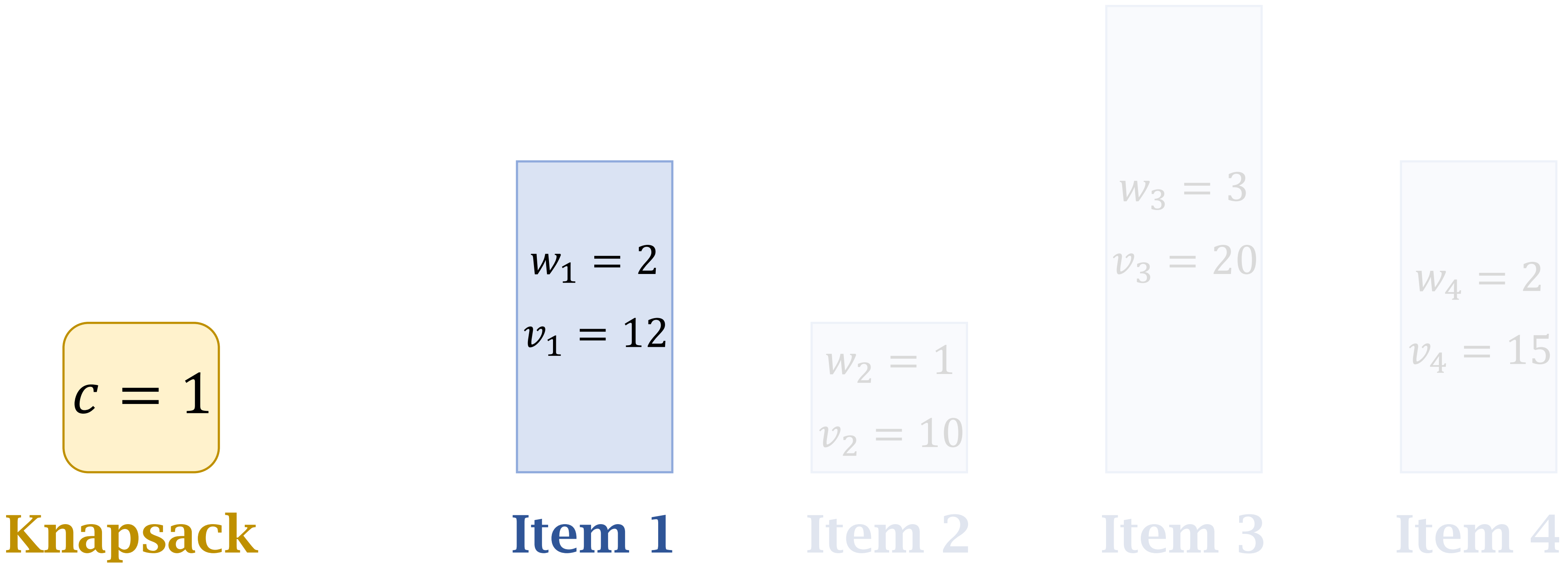
- F[m][c]: The maximum value of the subset of items that fit in the knapsack.
- F[4][5] $= 12 + 10 + 15 = 37$.





Sub-problem: m=1 and c=1

- Item 1 does not fit in the small knapsack. ($w_1 > c$.)
- $\boxed{F[1][1] = 0.}$



Sub-problem: m=1 and c=2

- Put Item 1 in the small knapsack. ($w_1 \leq c$.)
- $\boxed{\mathrm{F}[1][2] = v_1 = 12.}$

$c = 2$

Knapsack

$w_1 = 2$

$v_1 = 12$

Item 1

$w_2 = 1$

$v_2 = 10$

Item 2

$w_3 = 3$

$v_3 = 20$

Item 3

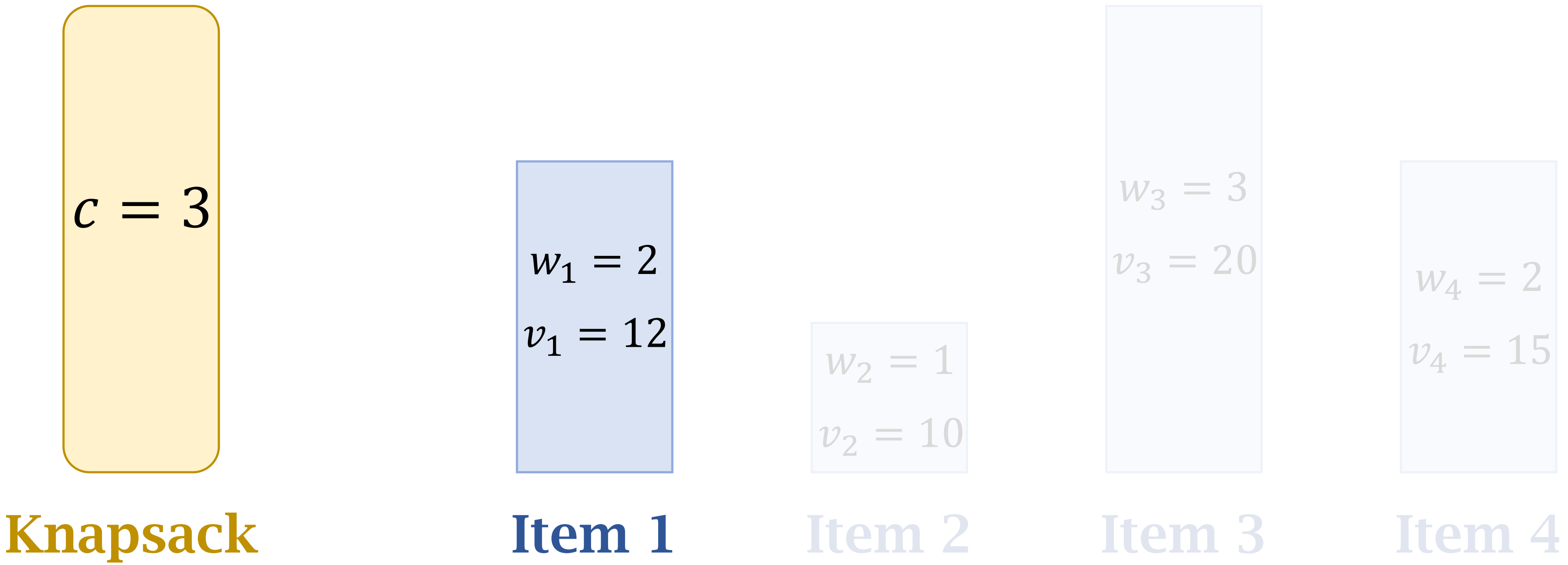
$w_4 = 2$

$v_4 = 15$

Item 4

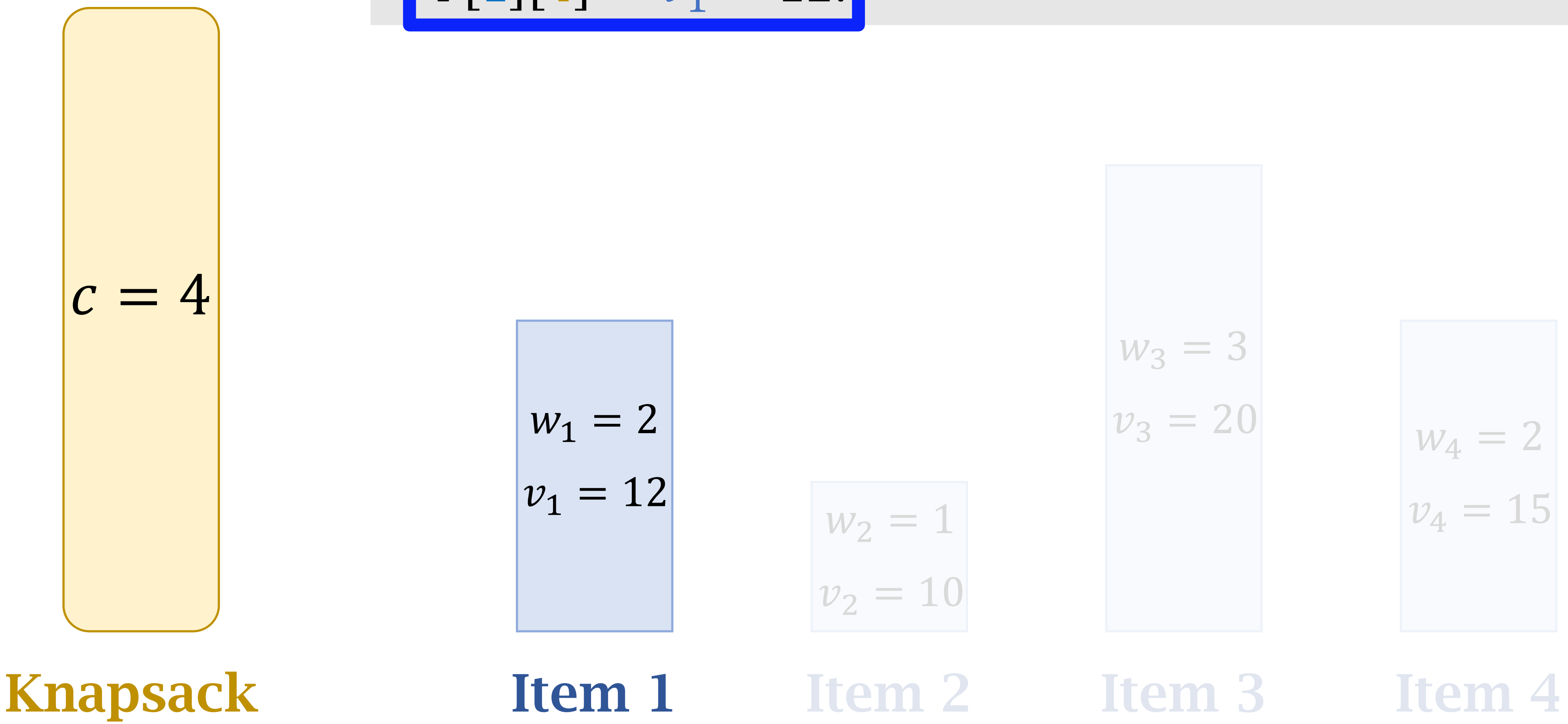
Sub-problem: m=1 and c=3

- Put Item 1 in the small knapsack. ($w_1 \leq c$.)
- $\boxed{\mathrm{F}[1][3] = v_1 = 12.}$



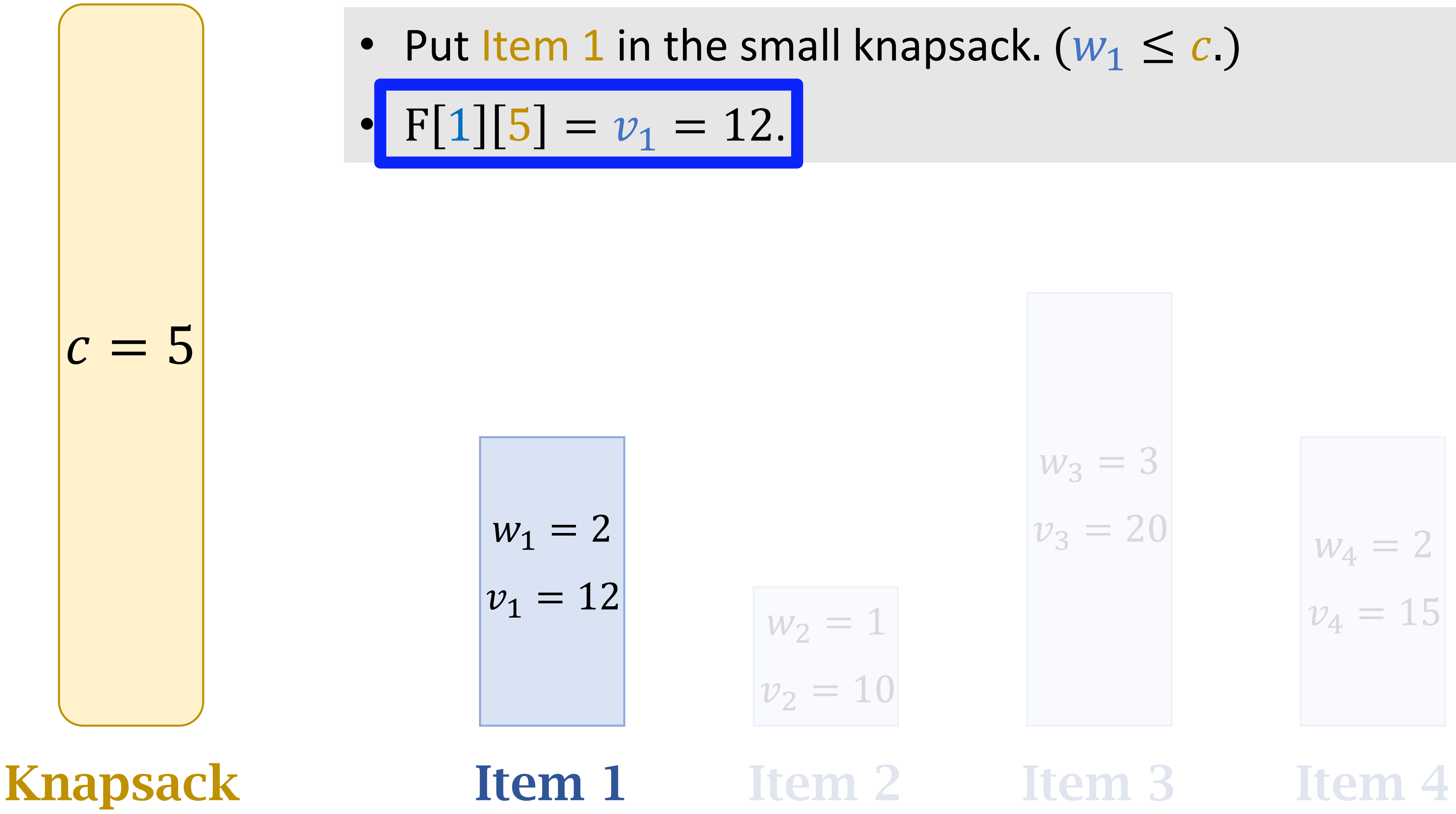
Sub-problem: m=1 and c=4

- Put Item 1 in the small knapsack. ($w_1 \leq c$.)
- $\boxed{F[1][4] = v_1 = 12.}$



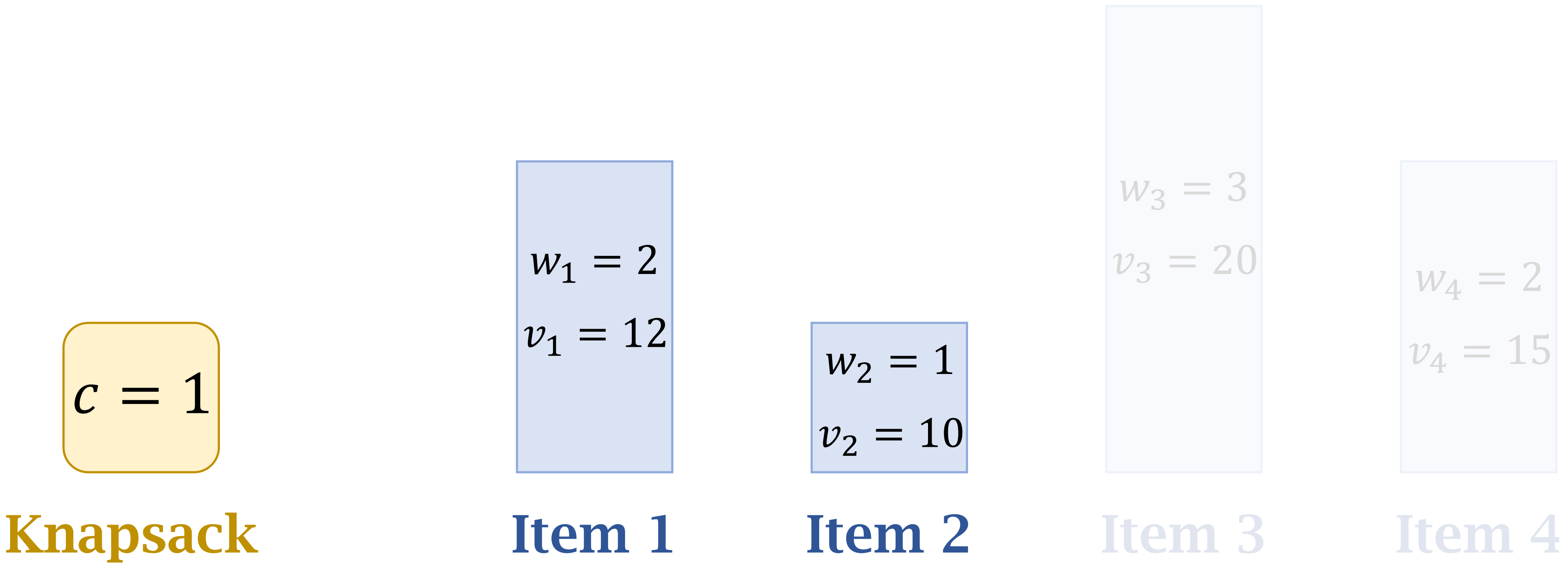
Sub-problem: m=1 and c=5

- Put Item 1 in the small knapsack. ($w_1 \leq c$.)
- $\boxed{F[1][5] = v_1 = 12.}$



Sub-problem: m=2 and c=1

- Put Item 2 in the knapsack. ($w_2 \leq c$.)
- $\boxed{\text{F}[2][1] = v_2 = 10.}$



Sub-problem: m=2 and c=2

- Put Item 1 in the knapsack. ($w_1 \leq c$.)
- $\boxed{F[2][2] = v_1 = 12.}$

$c = 2$

Knapsack

$w_1 = 2$

$v_1 = 12$

Item 1

$w_2 = 1$

$v_2 = 10$

Item 2

$w_3 = 3$

$v_3 = 20$

Item 3

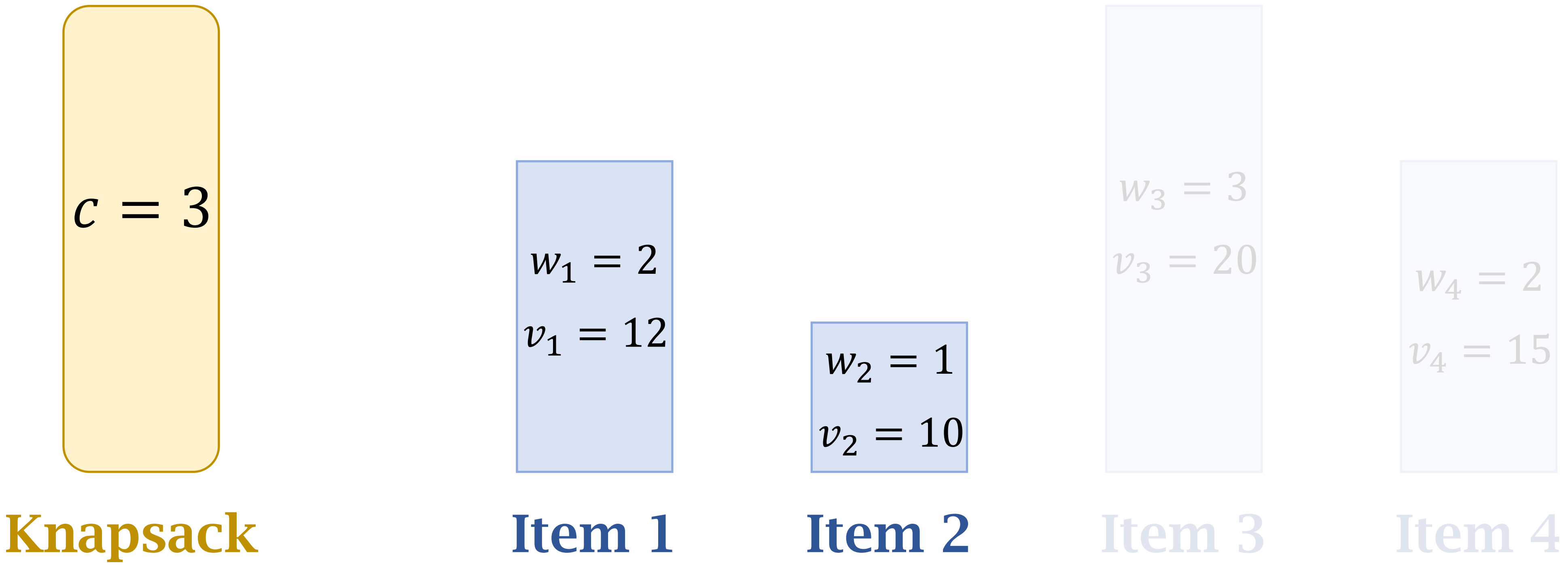
$w_4 = 2$

$v_4 = 15$

Item 4

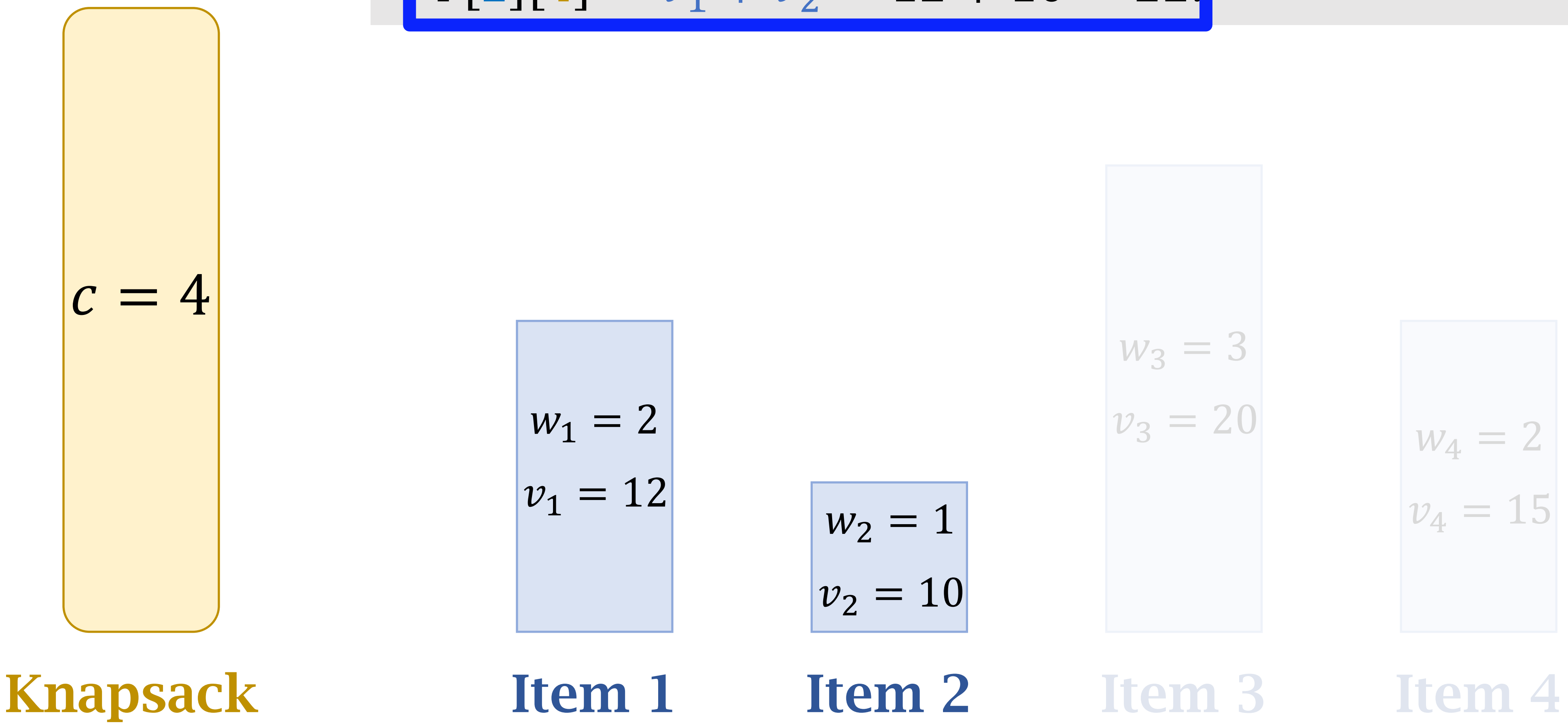
Sub-problem: m=2 and c=3

- Put both Items 1 and 2 in the knapsack. ($w_1 + w_2 \le c$.)
- $\boxed{F[2][3] = v_1 + v_2 = 12 + 10 = 22.}$



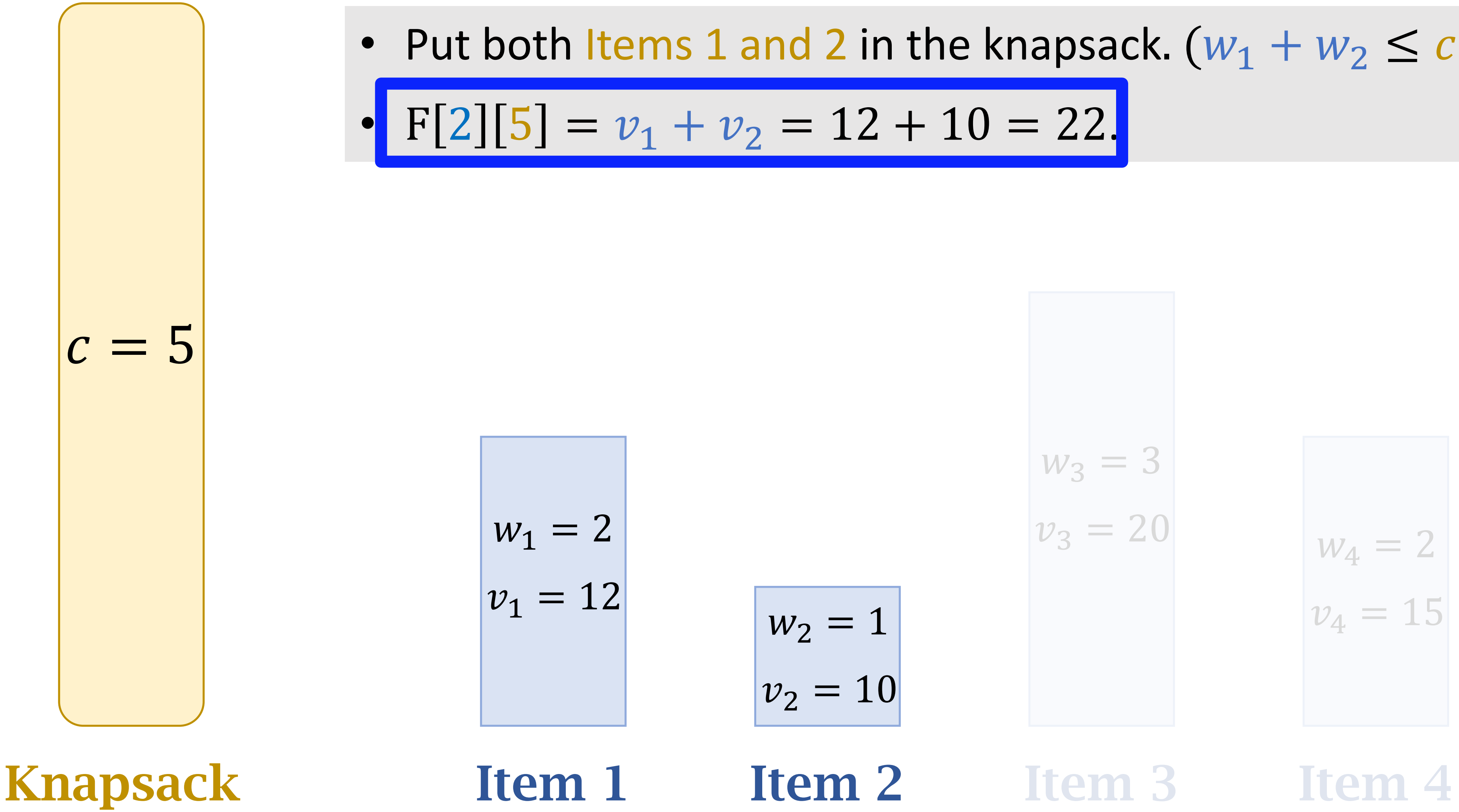
Sub-problem: m=2 and c=4

- Put both Items 1 and 2 in the knapsack. ($w_1 + w_2 \leq c$.)
- $\boxed{\mathrm{F}[2][4] = v_1 + v_2 = 12 + 10 = 22.}$



Sub-problem: m=2 and c=5

- Put both Items 1 and 2 in the knapsack. ($w_1 + w_2 \leq c$.)
- $\boxed{F[2][5] = v_1 + v_2 = 12 + 10 = 22.}$



Derive Optimal Substructure

- There are m items and a knapsack of capacity c.
- $\mathrm{F}[m][c]$: max value of the subset of items that fit in the knapsack.

Derive Optimal Substructure

- There are m items and a knapsack of capacity c.
- $\mathrm{F}[m][c]$: max value of the subset of items that fit in the knapsack.

There are two decisions for the m-th item: selecting it or not.

Derive Optimal Substructure

- There are m items and a knapsack of capacity c.
- $\mathrm{F}[m][c]$: max value of the subset of items that fit in the knapsack.

What if the m-th item is selected?

- Sub-problem:
  - The available capacity of the knapsack becomes $c - w_m$.
  - Only the first $m - 1$ items are available.
  - The sub-problem is to find $\mathrm{F}[m-1][c-w_m]$.

Derive Optimal Substructure

- There are m items and a knapsack of capacity c.
- $\mathrm{F}[m][c]$: max value of the subset of items that fit in the knapsack.

What if the m-th item is selected?

- Sub-problem:
  - The available capacity of the knapsack becomes $c - w_m$.
  - Only the first $m - 1$ items are available.
  - The sub-problem is to find $\mathrm{F}[m-1][c-w_m]$.

The maximum total value is: $v_m + \mathrm{F}[m-1][c-w_m]$.

Derive Optimal Substructure

- There are m items and a knapsack of capacity c.
- $\mathrm{F}[m][c]$: max value of the subset of items that fit in the knapsack.

What if the m-th item is not selected?

- Sub-problem:
  - The available capacity of the knapsack is still c.
  - Only the first $m-1$ items are available.
  - The sub-problem is to find $\mathrm{F}[m-1][c]$.

Derive Optimal Substructure

- There are m items and a knapsack of capacity c.
- $\mathrm{F}[m][c]$: max value of the subset of items that fit in the knapsack.

What if the m-th item is not selected?

- Sub-problem:
  - The available capacity of the knapsack is still c.
  - Only the first $m-1$ items are available.
  - The sub-problem is to find $\mathrm{F}[m-1][c]$.

The maximum total value is: $\mathrm{F}[m-1][c]$.

Case 1: $\boxed{w_m \leq c}$

- There are two decisions: selecting the m-th item or not.

Case 1: $w_m \leq c$

- There are two decisions: selecting the m-th item or not.
- If selected, then

$$\mathrm{F}[m][c] = v_m + \mathrm{F}[m-1][c-w_m].$$

- If not selected, then

$$\mathrm{F}[m][c] = \mathrm{F}[m-1][c].$$

Case 1: $w_m \le c$

- There are two decisions: selecting the m-th item or not.
- If selected, then

$$\mathrm{F}[m][c] = v_m + \mathrm{F}[m-1][c-w_m].$$

- If not selected, then

$$\mathrm{F}[m][c] = \mathrm{F}[m-1][c].$$

- In sum, $\mathrm{F}[m][c] = \max \begin{Bmatrix} v_m + \mathrm{F}[m-1][c-w_m] \\ \mathrm{F}[m-1][c] \end{Bmatrix}.$

Case 2: $w_m > c$

- The m-th item is too heavy to fit in the knapsack.
- The m-th item is not selected, and thus

$$\mathrm{F}[m][c] = \mathrm{F}[m-1][c].$$

Optimal Substructure

Case 1: $w_m \leq c$

- $\mathrm{F}[m][c] = \max\left\{\begin{matrix} v_m + \mathrm{F}[m-1][c-w_m] \\ \mathrm{F}[m-1][c] \end{matrix}\right\}.$

Case 2: $w_m > c$

- $\mathrm{F}[m][c] = \mathrm{F}[m-1][c].$

Dynamic Programming

Initialization

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Initialization

$F[m][c]$ is the (m, c)-th entry in the right table.

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5
0						
1						
2						
3						
4						

Base Case: `m = 0`

$F[0][c] = 0$, for all c, because no item can be selected.

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5
0						
1						
2						
3						
4						

Base Case: m = 0

$F[0][c] = 0$, for all c, because no item can be selected.

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5
0	0	0	0	0	0	0
1						
2						
3						
4						

Base Case: c = 0

$F[m][0] = 0$, for all m, because the knapsack is too small.

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5
0	0	0	0	0	0	0
1						
2						
3						
4						

Base Case: c = 0

$F[m][0] = 0$, for all m, because the knapsack is too small.

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0					
2	0					
3	0					
4	0					

$m = 1, \ c = 1$

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0					
2	0					
3	0					
4	0					

$m = 1, \; c = 1$

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	?				
2	0					
3	0					
4	0					

m = 1, c = 1

$$w_m > c \quad ==> \quad \boxed{\mathrm{F}[m][c] = \mathrm{F}[m-1][c]}.$$

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	?				
2	0					
3	0					
4	0					

$m = 1, \; c = 1$

$$\boxed{F[1][1] = F[0][1].}$$

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0				
2	0					
3	0					
4	0					

$m = 1, \; c = 4$

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0			?	
2	0					
3	0					
4	0					

$m = 1, \; c = 4$

$$w_m \leq c \quad ==> \quad \mathrm{F}[m][c] = \max \begin{Bmatrix} v_m + \mathrm{F}[m-1][c-w_m] \\ \mathrm{F}[m-1][c] \end{Bmatrix}.$$

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$

	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0			?	
2	0					
3	0					
4	0					

m = 1, c = 4

$$F[1][4] = \max\begin{cases} v_1 + F[0][2] \\ F[0][4] \end{cases} = \max\begin{cases} 12 + 0 \\ 0 \end{cases} = 12.$$

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$

	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0			12	
2	0					
3	0					
4	0					

m = 1

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	12	12	12	12
2	0					
3	0					
4	0					

$m = 2$

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	12	12	12	12
2	0					
3	0					
4	0					

`m = 2`

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	12	12	12	12
2	0					
3	0					
4	0					

`m = 2, c = 1`

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	12	12	12	12
2	0	?				
3	0					
4	0					

m = 2, c = 1

$$w_m \leq c \quad ==> \quad F[m][c] = \max\begin{Bmatrix} v_m + F[m-1][c-w_m] \\ F[m-1][c] \end{Bmatrix}.$$

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	12	12	12	12
2	0	?				
3	0					
4	0					

`m = 2, c = 1`

$$F[2][1] = \max\begin{Bmatrix} v_2 + F[1][0] \\ F[1][1] \end{Bmatrix} = \max\begin{Bmatrix} 10 + 0 \\ 0 \end{Bmatrix} = 10.$$

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	12	12	12	12
2	0	10				
3	0					
4	0					

$$m = 2, \quad c = 3$$

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	12	12	12	12
2	0	10		?		
3	0					
4	0					

$\mathtt{m = 2,\ c = 3}$

$$w_m \le c \quad ==> \quad F[m][c] = \max\left\{\begin{matrix} v_m + F[m-1][c-w_m] \\ F[m-1][c] \end{matrix}\right\}.$$

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	12	12	12	12
2	0	10		?		
3	0					
4	0					

m = 2, c = 3

$$F[2][3] = \max\begin{Bmatrix} v_2 + F[1][2] \\ F[1][3] \end{Bmatrix} = \max\begin{Bmatrix} 10 + 12 \\ 12 \end{Bmatrix} = 22.$$

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	12	12	12	12
2	0	10		22		
3	0					
4	0					

m = 2

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	12	12	12	12
2	0	10	12	22	22	22
3	0					
4	0					

$\mathtt{m} = 3$

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	12	12	12	12
2	0	10	12	22	22	22
3	0					
4	0					

$m = 3$

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	12	12	12	12
2	0	10	12	22	22	22
3	0					
4	0					

$m = 3, \; c = 2$

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	12	12	12	12
2	0	10	12	22	22	22
3	0		?			
4	0					

$m = 3, \; c = 2$

$w_m > c$ ==> $F[m][c] = F[m-1][c].$

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$

	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	12	12	12	12
2	0	10	12	22	22	22
3	0		?			
4	0					

m = 3, c = 2

$$F[3][2] = F[2][2].$$

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	12	12	12	12
2	0	10	12	22	22	22
3	0		12			
4	0					

$$m = 3, \quad c = 4$$

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$

	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	12	12	12	12
2	0	10	12	22	22	22
3	0		12		?	
4	0					

`m = 3, c = 4`

$$w_m \le c \quad \text{==>} \quad \mathrm{F}[m][c] = \max\left\{\begin{matrix} v_m + \mathrm{F}[m-1][c-w_m] \\ \mathrm{F}[m-1][c] \end{matrix}\right\}.$$

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

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	12	12	12	12
2	0	10	12	22	22	22
3	0		12		?	
4	0					

m = 3, c = 4

$$\mathrm{F}[3][4] = \max\begin{Bmatrix} v_3 + \mathrm{F}[2][1] \\ \mathrm{F}[2][4] \end{Bmatrix} = \max\begin{Bmatrix} 20 + 10 \\ 22 \end{Bmatrix} = 30.$$

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	12	12	12	12
2	0	10	12	22	22	22
3	0		12		30	
4	0					

m = 3

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	12	12	12	12
2	0	10	12	22	22	22
3	0	10	12	22	30	32
4	0					

m = 4

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	12	12	12	12
2	0	10	12	22	22	22
3	0	10	12	22	30	32
4	0					

$m = 4, \ c = 5$

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$

	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	12	12	12	12
2	0	10	12	22	22	22
3	0	10	12	22	30	32
4	0					?

$$\mathbf{m = 4,\ c = 5}$$

$$w_m \le c \quad ==> \quad F[m][c] = \max\left\{\begin{matrix} v_m + F[m-1][c-w_m] \\ F[m-1][c] \end{matrix}\right\}.$$

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$

	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	12	12	12	12
2	0	10	12	22	22	22
3	0	10	12	22	30	32
4	0					?

m = 4, c = 5

$$F[4][5] = \max\begin{Bmatrix} v_4 + F[3][3] \\ F[3][5] \end{Bmatrix} = \max\begin{Bmatrix} 15 + 22 \\ 32 \end{Bmatrix} = 37.$$

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	12	12	12	12
2	0	10	12	22	22	22
3	0	10	12	22	30	32
4	0					37

Find the selected items

Find the selected items

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	12	12	12	12
2	0	10	12	22	22	22
3	0	10	12	22	30	32
4	0					37

Is the 4th item selected?

- 37 ≠ 32 ==> Item 4 is selected.

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	12	12	12	12
2	0	10	12	22	22	22
3	0	10	12	22	30	32
4	0					37

The 4^{th} item is selected!

- $37 \neq 32$ ==> Item 4 is selected.
- The remaining capacity is $5 - 2 = 3$.

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

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	12	12	12	12
2	0	10	12	22	22	22
3	0	10	12	22	30	32
4	0					37

The 4th item is selected!

- $37 \neq 32$ ==> Item 4 is selected.
- The remaining capacity is $5 - 2 = 3$.

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$	0	1	2	$\boxed{3}$	4	5
0	0	0	0	0	0	0
1	0	0	12	12	12	12
2	0	10	12	22	22	22
3	0	10	12	22	30	32
4	0					37

Is the 3rd item selected?

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	12	12	12	12
2	0	10	12	22	22	22
3	0	10	12	22	30	32
4	0					37

Is the 3rd item selected?

- $22 = 22$ ==> Item 3 is not selected.

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	12	12	12	12
2	0	10	12	22	22	22
3	0	10	12	22	30	32
4	0					37

The 3rd item is not selected!

- 22 $=$ 22 ==> Item 3 is not selected.
- The remaining capacity is still 3.

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	12	12	12	12
2	0	10	12	22	22	22
3	0	10	12	22	30	32
4	0					37

Is the 2nd item selected?

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

\# of Items $m =$

	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	12	12	12	12
2	0	10	12	22	22	22
3	0	10	12	22	30	32
4	0					37

Is the 2nd item selected?

- 22 $\neq$ 12 ==> Item 2 is selected.

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	12	12	12	12
2	0	10	12	22	22	22
3	0	10	12	22	30	32
4	0					37

The 2nd item is selected!

- $22 \neq 12$ ==> Item 2 is selected.
- The remaining capacity is $3 - 1 = 2$.

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

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	12	12	12	12
2	0	10	12	22	22	22
3	0	10	12	22	30	32
4	0					37

The 2nd item is selected!

- 22 $\neq$ 12 ==> Item 2 is selected.
- The remaining capacity is $3 - 1 = 2$.

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	12	12	12	12
2	0	10	12	22	22	22
3	0	10	12	22	30	32
4	0					37

Is the 1st item selected?

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	12	12	12	12
2	0	10	12	22	22	22
3	0	10	12	22	30	32
4	0					37

Is the 1st item selected?

- $12 \neq 0$ ==> Item 1 is selected.

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$

	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	12	12	12	12
2	0	10	12	22	22	22
3	0	10	12	22	30	32
4	0					37

Is the 1st item selected?

- $12 \neq 0$ ==> Item 1 is selected.

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	12	12	12	12
2	0	10	12	22	22	22
3	0	10	12	22	30	32
4	0					37

Found the selected items!

Items 1, 2, and 4 are selected.

Item	Weight w_m	Value v_m
1	2	12
2	1	10
3	3	20
4	2	15

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	12	12	12	12
2	0	10	12	22	22	22
3	0	10	12	22	30	32
4	0					37

Time Complexity

Time Complexity

- Select from m items to fit in a knapsack of capacity c.
- The size of table is $(m+1)\times(c+1)$.
- $O(1)$ time for filling one entry of the table.
- Thus, $O(mc)$ time for filling the entire table.

Time Complexity

- Properties of the items can be represented by $2m$ integers.
  - Let $x = \max\{w_1, \cdots, w_m, v_1, \cdots, v_m\}$.
  - Each integer is represented by $\log_2 x$ bits.
  - Number of bits: $b = 2m \cdot \log_2 x$.

Time Complexity

- Properties of the items can be represented by $2m$ integers.
  - Let $x = \max\{w_1, \cdots, w_m, v_1, \cdots, v_m\}$.
  - Each integer is represented by $\log_2 x$ bits.
  - Number of bits: $b = 2m \cdot \log_2 x$.
- Capacity of the knapsack can be represented by $k = \log_2 c$ bits.
- The time complexity is

$$\underline{O(mc)} = O\underline{\left(\frac{b}{2 \cdot \log_2 x} \cdot 2^k\right)} = \underline{O(b \cdot 2^k)}.$$

Questions

Q1: Fill the table

Item	Weight w_m	Value v_m
1	5	50
2	4	40
3	4	30
4	3	45

Capacity $c =$

# of Items $m =$

m \ c	0	1	2	3	4	5	6	7	8	9	10
0											
1											
2											
3											
4											

Q2: Fill the table

Item	Weight w_m	Value v_m
1	1	5
2	3	35
3	4	38
4	5	60
5	2	20

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5	6	7	8	9	10
0											
1											
2											
3											
4											
5											

Q3: What are the selected items?

Item	Weight w_m	Value v_m
1	1	
2	3	
3	4	
4	5	
5	2	

Capacity $c =$

# of Items $m =$	0	1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	0	0	0
1	0	5	5	5	5	5	5	5	5	5	5
2	0	5	5	35	40	40	40	40	40	40	40
3	0	5	5	35	40	40	40	65	70	70	70
4	0	5	5	35	40	55	60	65	90	95	95
5	0	5	22	35	40	57	62	77	90	95	112

Thank You!