



Seminario: Programación dinámica

© Alberto Valderruten, Óscar Fontenla Romero

{Alberto.valderruten, oscar.fontenla}@udc.es

Devolver el cambio

Ejercicio 1: construir la tabla con la que podría determinarse en programación dinámica la manera óptima de pagar una cantidad de 17 unidades de valor con un mínimo de monedas, sabiendo que el sistema monetario considerado está constituido por monedas de 1, 3, 8 y 12 unidades de valor. Indicar la solución al problema dibujando una traza en la tabla anterior para justificar cómo se obtiene.

Devolver el cambio

Denominación
de la moneda

Importe a pagar

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1																		
3																		
8																		
12																		

Cada celda contendrá el número mínimo de monedas
necesarias para pagar el importe indicado por la columna y
considerando las denominaciones desde la 1 hasta la fila dada

Devolver el cambio

Denominación
de la moneda

Importe a pagar

[illegible]

Devolver el cambio

Denominación
de la moneda

Importe a pagar

[illegible]

Devolver el cambio

Denominación
de la moneda

Importe a pagar

[illegible]

Devolver el cambio

Denominación
de la moneda

Importe a pagar

[illegible]

Devolver el cambio

Denominación
de la moneda

Importe a pagar

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
3	0	1	2															
8	0																	
12	0																	

Diagram illustrating a jump table structure. The table has 18 columns (0 to 17) and 4 rows (1 to 12). The first row (index 1) contains values 0 through 17. The second row (index 3) contains values 0 through 2, followed by a jump of -1 (Salto: -1) to the value 3 in the first row. The third row (index 8) contains the value 0, followed by a jump of -3 (Salto: -3) to the value 0 in the first row. The fourth row (index 12) contains the value 0.

Devolver el cambio

Denominación
de la moneda

Importe a pagar

[illegible]

Devolver el cambio

Denominación
de la moneda

Importe a pagar

[illegible]

Devolver el cambio

Denominación
de la moneda

Importe a pagar

[illegible]

Devolver el cambio

Denominación
de la moneda

Importe a pagar

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
3	0	1	2	1														
8	0																	
12	0																	

Diagram illustrating the change-making problem. The table shows the number of coins (0 to 17) for each denomination (0 to 17). The path highlighted shows the sequence of coins used to make the change, with jumps labeled "Salto: -1" and "Salto: -3".

Devolver el cambio

Denominación
de la moneda

Importe a pagar

[illegible]

Devolver el cambio

Denominación
de la moneda

Importe a pagar

[illegible]

Devolver el cambio

Denominación
de la moneda

Importe a pagar

[illegible]

Devolver el cambio

Denominación
de la moneda

Importe a pagar

[illegible]

Devolver el cambio

Denominación
de la moneda

Importe a pagar

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
3	0	1	2	1	2													
8	0																	
12	0																	

Diagram illustrating a jump table structure. The table has 18 columns (0 to 17) and 4 rows (1 to 12). The first row (1) contains values 0 to 17. The second row (3) contains values 0 to 17, with a jump of -1 from column 5 to column 6. The third row (8) contains values 0 to 17, with a jump of -3 from column 5 to column 2. The fourth row (12) contains values 0 to 17.

Devolver el cambio

Denominación
de la moneda

Importe a pagar

[illegible]

Devolver el cambio

Denominación
de la moneda

Importe a pagar

[illegible]

Devolver el cambio

Denominación
de la moneda

Importe a pagar

[illegible]

Devolver el cambio

Denominación
de la moneda

Importe a pagar

[illegible]

Devolver el cambio

Denominación
de la moneda

Importe a pagar

[illegible]

Devolver el cambio

Denominación
de la moneda

Importe a pagar

[illegible]

Devolver el cambio

Denominación
de la moneda

Importe a pagar

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
3	0	1	2	1	2	3	2	3	4	3	4	5	4	5	6	5	6	7
8	0	1	2	1	2	3	2	3										
12	0																	

Diagram illustrating a memory access pattern for a 2D array (rows 1, 3, 8, 12; columns 0-17). The array is divided into four groups of 8 columns each, indicated by a bracket at the top.

Arrows indicate the sequence of memory accesses:

- From row 1, column 8 to row 3, column 4 (labeled **Salto: -1**).
- From row 3, column 4 to row 8, column 0 (labeled **Salto: -8**).

Devolver el cambio

Denominación
de la moneda

Importe a pagar

[illegible]

Devolver el cambio

Denominación
de la moneda

Importe a pagar

[illegible]

Denominación
de la moneda

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
2	0	1	2	1	2	3	2	3	4	3	4	5	4	5	6	5	6	7
3	0	1	2	1	2	3	2	3	1									

Salto: -1

Salto: -8

Devolver el cambio

Denominación
de la moneda

Importe a pagar

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
3	0	1	2	1	2	3	2	3	4	3	4	5	4	5	6	5	6	7
8	0	1	2	1	2	3	2	3	1	2								
12	0																	

Salto: -8 (from index 12 to index 4)

Salto: -1 (from index 9 to index 10)

Devolver el cambio

Denominación
de la moneda

Importe a pagar

[illegible]

Devolver el cambio

Denominación
de la moneda

Importe a pagar

[illegible]

Devolver el cambio

Denominación
de la moneda

Importe a pagar

The diagram illustrates the change-making problem. It features a grid where the columns represent the amount to be paid (0 to 17) and the rows represent the number of coins of a specific denomination (1, 3, 8, 12). The top row of the grid shows the denominations 0 through 17. The first column shows the denominations 1, 3, 8, and 12. The grid cells contain the number of coins of that denomination needed to make the amount. For example, to make 17 with 12-cent coins, you need 1 coin of 12 and 5 coins of 1. Curved arrows in the 8 and 12 rows indicate the transition from one state to the next by adding a coin of the respective denomination.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
3	0	1	2	1	2	3	2	3	4	3	4	5	4	5	6	5	6	7
8	0	1	2	1	2	3	2	3	1	2	3	2	3	4	3	4	2	3
12	0	1	2	1	2	3	2	3	1	2	3	2						

Devolver el cambio

Denominación
de la moneda

Importe a pagar

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
3	0	1	2	1	2	3	2	3	4	3	4	5	4	5	6	5	6	7
8	0	1	2	1	2	3	2	3	1	2	3	2	3	4	3	4	2	3
12	0	1	2	1	2	3	2	3	1	2	3	2						

Devolver el cambio

Denominación
de la moneda

Importe a pagar

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
3	0	1	2	1	2	3	2	3	4	3	4	5	4	5	6	5	6	7
8	0	1	2	1	2	3	2	3	1	2	3	2	3	4	3	4	2	3
12	0	1	2	1	2	3	2	3	1	2	3	2		Salto: -1				

Salto: -12

Devolver el cambio

Denominación
de la moneda

Importe a pagar

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
3	0	1	2	1	2	3	2	3	4	3	4	5	4	5	6	5	6	7
8	0	1	2	1	2	3	2	3	1	2	3	2	3	4	3	4	2	3
12	0	1	2	1	2	3	2	3	1	2	3	2	1					

Salto: -12

Salto: -1

Devolver el cambio

Denominación
de la moneda

Importe a pagar



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
3	0	1	2	1	2	3	2	3	4	3	4	5	4	5	6	5	6	7
8	0	1	2	1	2	3	2	3	1	2	3	2	3	4	3	4	2	3
12	0	1	2	1	2	3	2	3	1	2	3	2	1					

Devolver el cambio

Denominación
de la moneda

Importe a pagar



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
3	0	1	2	1	2	3	2	3	4	3	4	5	4	5	6	5	6	7
8	0	1	2	1	2	3	2	3	1	2	3	2	3	4	3	4	2	3
12	0	1	2	1	2	3	2	3	1	2	3	2	1	2	3	2	2	3

Devolver el cambio

Denominación
de la moneda

Importe a pagar

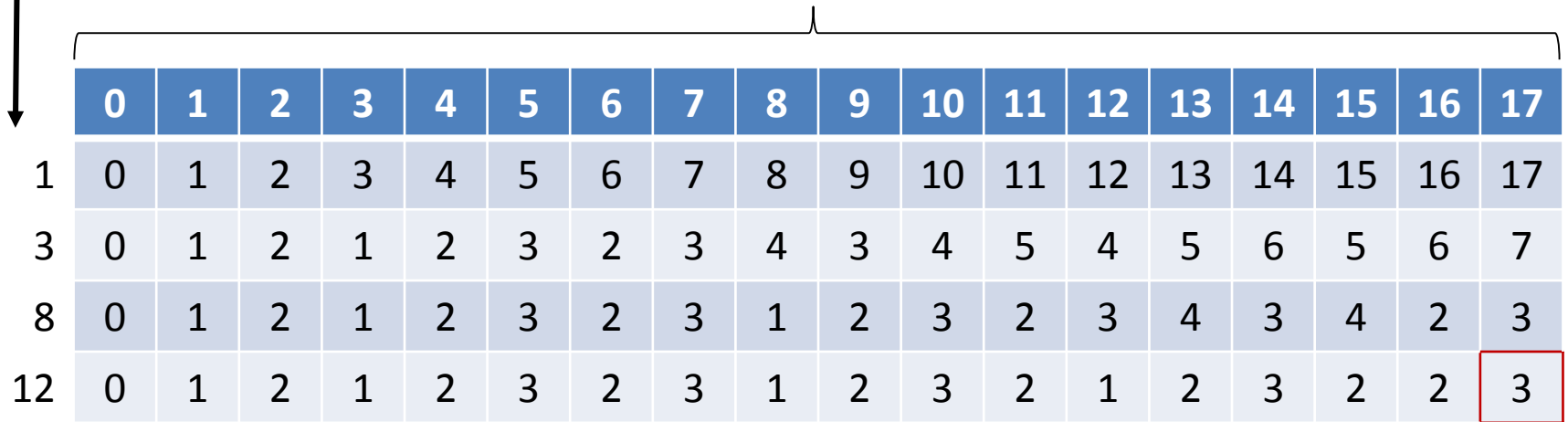
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
3	0	1	2	1	2	3	2	3	4	3	4	5	4	5	6	5	6	7
8	0	1	2	1	2	3	2	3	1	2	3	2	3	4	3	4	2	3
12	0	1	2	1	2	3	2	3	1	2	3	2	1	2	3	2	2	3

Número óptimo de monedas
para pagar una cantidad de 17

Devolver el cambio

Denominación
de la moneda

Importe a pagar



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
3	0	1	2	1	2	3	2	3	4	3	4	5	4	5	6	5	6	7
8	0	1	2	1	2	3	2	3	1	2	3	2	3	4	3	4	2	3
12	0	1	2	1	2	3	2	3	1	2	3	2	1	2	3	2	2	3

¿Cuáles son esas monedas?:
Construir la traza desde este
elemento ($c[m,n]$) hasta $c[0,0]$

Devolver el cambio

Denominación
de la moneda

Importe a pagar

	Importe a pagar																	
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
3	0	1	2	1	2	3	2	3	4	3	4	5	4	5	6	5	6	7
8	0	1	2	1	2	3	2	3	1	2	3	2	3	4	3	4	2	3
12	0	1	2	1	2	3	2	3	1	2	3	2	1	2	3	2	2	3

No usar más
monedas de 12

Devolver el cambio

Denominación
de la moneda

Importe a pagar



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
3	0	1	2	1	2	3	2	3	4	3	4	5	4	5	6	5	6	7
8	0	1	2	1	2	3	2	3	1	2	3	2	3	4	3	4	2	3
12	0	1	2	1	2	3	2	3	1	2	3	2	1	2	3	2	2	3

Devolver el cambio

Denominación
de la moneda

Importe a pagar

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
3	0	1	2	1	2	3	2	3	4	3	4	5	4	5	6	5	6	7
8	0	1	2	1	2	3	2	3	1	2	3	2	3	4	3	4	2	3
12	0	1	2	1	2	3	2	3	1	2	3	2	1	2	3	2	2	3

8

Devolver el cambio

Denominación
de la moneda

Importe a pagar

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
3	0	1	2	1	2	3	2	3	4	3	4	5	4	5	6	5	6	7
8	0	1	2	1	2	3	2	3	1	2	3	2	3	4	3	4	2	3
12	0	1	2	1	2	3	2	3	1	2	3	2	1	2	3	2	2	3

Salto: -8

Devolver el cambio

Denominación
de la moneda

Importe a pagar

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
3	0	1	2	1	2	3	2	3	4	3	4	5	4	5	6	5	6	7
8	0	1	2	1	2	3	2	3	1	2	3	2	3	4	3	4	2	3
12	0	1	2	1	2	3	2	3	1	2	3	2	1	2	3	2	2	3

Devolver el cambio

Denominación
de la moneda

Importe a pagar

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
3	0	1	2	1	2	3	2	3	4	3	4	5	4	5	6	5	6	7
8	0	1	2	1	2	3	2	3	1	2	3	2	3	4	3	4	2	3
12	0	1	2	1	2	3	2	3	1	2	3	2	1	2	3	2	2	3



Devolver el cambio

Denominación
de la moneda

Importe a pagar

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
3	0	1	2	1	2	3	2	3	4	3	4	5	4	5	6	5	6	7
8	0	1	2	1	2	3	2	3	1	2	3	2	3	4	3	4	2	3
12	0	1	2	1	2	3	2	3	1	2	3	2	1	2	3	2	2	3

Salto: -8

8

8

Devolver el cambio

Denominación
de la moneda

Importe a pagar

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
3	0	1	2	1	2	3	2	3	4	3	4	5	4	5	6	5	6	7
8	0	1	2	1	2	3	2	3	1	2	3	2	3	4	3	4	2	3
12	0	1	2	1	2	3	2	3	1	2	3	2	1	2	3	2	2	3



Devolver el cambio

Denominación
de la moneda

Importe a pagar

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
3	0	1	2	1	2	3	2	3	4	3	4	5	4	5	6	5	6	7
8	0	1	2	1	2	3	2	3	1	2	3	2	3	4	3	4	2	3
12	0	1	2	1	2	3	2	3	1	2	3	2	1	2	3	2	2	3

No usar más
monedas de 8



Devolver el cambio

Denominación
de la moneda

Importe a pagar

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
3	0	1	2	1	2	3	2	3	4	3	4	5	4	5	6	5	6	7
8	0	1	2	1	2	3	2	3	1	2	3	2	3	4	3	4	2	3
12	0	1	2	1	2	3	2	3	1	2	3	2	1	2	3	2	2	3



Devolver el cambio

Denominación
de la moneda

Importe a pagar

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
3	0	1	2	1	2	3	2	3	4	3	4	5	4	5	6	5	6	7
8	0	1	2	1	2	3	2	3	1	2	3	2	3	4	3	4	2	3
12	0	1	2	1	2	3	2	3	1	2	3	2	1	2	3	2	2	3

No usar más
monedas de 3



Devolver el cambio

Denominación
de la moneda

Importe a pagar

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
3	0	1	2	1	2	3	2	3	4	3	4	5	4	5	6	5	6	7
8	0	1	2	1	2	3	2	3	1	2	3	2	3	4	3	4	2	3
12	0	1	2	1	2	3	2	3	1	2	3	2	1	2	3	2	2	3



Devolver el cambio

Denominación
de la moneda

Importe a pagar

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
3	0	1	2	1	2	3	2	3	4	3	4	5	4	5	6	5	6	7
8	0	1	2	1	2	3	2	3	1	2	3	2	3	4	3	4	2	3
12	0	1	2	1	2	3	2	3	1	2	3	2	1	2	3	2	2	3



Devolver el cambio

Denominación
de la moneda

Importe a pagar

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
3	0	1	2	1	2	3	2	3	4	3	4	5	4	5	6	5	6	7
8	0	1	2	1	2	3	2	3	1	2	3	2	3	4	3	4	2	3
12	0	1	2	1	2	3	2	3	1	2	3	2	1	2	3	2	2	3

Salto: -1



Devolver el cambio

Denominación
de la moneda

Importe a pagar

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
3	0	1	2	1	2	3	2	3	4	3	4	5	4	5	6	5	6	7
8	0	1	2	1	2	3	2	3	1	2	3	2	3	4	3	4	2	3
12	0	1	2	1	2	3	2	3	1	2	3	2	1	2	3	2	2	3



Devolver el cambio

Denominación
de la moneda

Importe a pagar

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
3	0	1	2	1	2	3	2	3	4	3	4	5	4	5	6	5	6	7
8	0	1	2	1	2	3	2	3	1	2	3	2	3	4	3	4	2	3
12	0	1	2	1	2	3	2	3	1	2	3	2	1	2	3	2	2	3



¿Por qué se descartaría el uso de la técnica voraz para resolver este problema?

El problema de la mochila

Ejercicio 2: se desea configurar la carga más valiosa posible para una mochila de capacidad limitada en peso (W), a partir de n objetos caracterizados por su peso (w_i) y su valor (v_i), ambos estrictamente positivos. Suponiendo que los objetos no se pueden fraccionar y que se dispone de una mochila de capacidad $W = 15$ unidades de peso, con el conjunto de objetos siguiente:

Objeto	1	2	3	4	5
v	3	4	6	5	5
w	2	3	4	5	6


- Construir la tabla con la que se podría encontrar mediante la técnica de Programación Dinámica la carga más valiosa posible para esta mochila, e indique en ella dos recorridos que correspondan a dos soluciones.
- Justificar la solución que daría la función voraz.

El problema de la mochila

Objeto

Capacidad de la mochila

	$\langle v, w \rangle$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	$\langle 3, 2 \rangle$																
2	$\langle 4, 3 \rangle$																
3	$\langle 6, 4 \rangle$																
4	$\langle 5, 5 \rangle$																
5	$\langle 5, 6 \rangle$																



Cada celda contendrá el valor máximo de la carga para la capacidad indicada por la columna y considerando los objetos desde 1 hasta la fila dada

El problema de la mochila

[illegible]

El problema de la mochila

[illegible]

El problema de la mochila

[illegible]

El problema de la mochila

[illegible]

El problema de la mochila

Objeto		Capacidad de la mochila															
	$\langle v, w \rangle$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	$\langle 3, 2 \rangle$	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	$\langle 4, 3 \rangle$	0	0	3													
3	$\langle 6, 4 \rangle$	0															
4	$\langle 5, 5 \rangle$	0															
5	$\langle 5, 6 \rangle$	0															

Diagram illustrating the knapsack problem. The table shows the maximum value v for each capacity w (0 to 15) considering the first i objects. The first row (Object 1) shows values for $\langle 3, 2 \rangle$. The second row (Object 2) shows values for $\langle 4, 3 \rangle$. The third row (Object 3) shows values for $\langle 6, 4 \rangle$. The fourth row (Object 4) shows values for $\langle 5, 5 \rangle$. The fifth row (Object 5) shows values for $\langle 5, 6 \rangle$. Red arrows and text indicate the transition from the first row to the second row, showing a jump of -1 in the value (Salto: -1) and a jump of -3 in the weight (Salto: -3 (w_2)).

El problema de la mochila

Objeto

Capacidad de la mochila

	<v, w>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	<3, 2>	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	<4, 3>	0	0	3	max												
3	<6, 4>	0		+v ₂													
4	<5, 5>	0															
5	<5, 6>	0															

El problema de la mochila

[illegible]

El problema de la mochila

[illegible]

El problema de la mochila

Objeto

Capacidad de la mochila

	<v, w>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	<3, 2>	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	<4, 3>	0	0	3	4												
3	<6, 4>	0															
4	<5, 5>	0															
5	<5, 6>	0															

Salto: -1

Salto: -3 (w_2)

El problema de la mochila

Objeto

Capacidad de la mochila

	<v, w>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	<3, 2>	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	<4, 3>	0	0	3	4	max											
3	<6, 4>	0															
4	<5, 5>	0															
5	<5, 6>	0															

El problema de la mochila

Objeto

Capacidad de la mochila

	<v, w>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	<3, 2>	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	<4, 3>	0	0	3	4	4											
3	<6, 4>	0															
4	<5, 5>	0															
5	<5, 6>	0															

Red arrows indicate the calculation for the value in the cell (2, 4):

- From the cell (1, 1) with value 0.
- From the cell (2, 3) with value 4.
- The result is 4, which is the value in the cell (2, 4).

The label $+v_2$ is placed near the arrow from (2, 3) to (2, 4).

El problema de la mochila

[illegible]

El problema de la mochila

Objeto

Capacidad de la mochila

	<v, w>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	<3, 2>	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	<4, 3>	0	0	3	4	4											
3	<6, 4>	0															
4	<5, 5>	0															
5	<5, 6>	0															

Salto: -1

Salto: -3 (w_2)

El problema de la mochila

Objeto

Capacidad de la mochila

	<v, w>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	<3, 2>	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	<4, 3>	0	0	3	4	4	max										
3	<6, 4>	0															
4	<5, 5>	0															
5	<5, 6>	0															

Diagram illustrating the knapsack problem. The table shows the maximum value for each object (1 to 5) across different capacities (0 to 15). Red arrows highlight the calculation for object 2 at capacity 5, showing the value 4 is the maximum of the previous value (3) and the value of object 2 plus the value of the remaining capacity (4 + 3 = 7). The cell is labeled 'max'.

El problema de la mochila

Objeto

Capacidad de la mochila

	<v, w>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	<3, 2>	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	<4, 3>	0	0	3	4	4	7										
3	<6, 4>	0															
4	<5, 5>	0															
5	<5, 6>	0															

El problema de la mochila

[illegible]

El problema de la mochila

[illegible]

El problema de la mochila

[illegible]

El problema de la mochila

[illegible]

El problema de la mochila

Objeto

Capacidad de la mochila

	<v, w>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	<3, 2>	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	<4, 3>	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	<6, 4>	0	0	3	4												
4	<5, 5>	0															
5	<5, 6>	0															

Salto: -1

Salto: -4 (w_3)

El problema de la mochila

Objeto		Capacidad de la mochila															
	$\langle v, w \rangle$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	$\langle 3, 2 \rangle$	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	$\langle 4, 3 \rangle$	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	$\langle 6, 4 \rangle$	0	0	3	4	max											
4	$\langle 5, 5 \rangle$	0															
5	$\langle 5, 6 \rangle$	0															

El problema de la mochila

Objeto		Capacidad de la mochila															
	$\langle v, w \rangle$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	$\langle 3, 2 \rangle$	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	$\langle 4, 3 \rangle$	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	$\langle 6, 4 \rangle$	0	0	3	4	6											
4	$\langle 5, 5 \rangle$	0															
5	$\langle 5, 6 \rangle$	0															

El problema de la mochila

[illegible]

El problema de la mochila

Objeto

Capacidad de la mochila

	<v, w>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	<3, 2>	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	<4, 3>	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	<6, 4>	0	0	3	4	6											
4	<5, 5>	0															
5	<5, 6>	0															

Salto: -1

Salto: -4 (w_3)

El problema de la mochila

Objeto

Capacidad de la mochila

	<v, w>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	<3, 2>	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	<4, 3>	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	<6, 4>	0	0	3	4	6	max										
4	<5, 5>	0															
5	<5, 6>	0															

El problema de la mochila

Objeto

Capacidad de la mochila

	<v, w>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	<3, 2>	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	<4, 3>	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	<6, 4>	0	0	3	4	6	7										
4	<5, 5>	0															
5	<5, 6>	0															

El problema de la mochila

[illegible]

El problema de la mochila

[illegible]

El problema de la mochila

[illegible]

El problema de la mochila

[illegible]

El problema de la mochila

Objeto		Capacidad de la mochila															
	$\langle v, w \rangle$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	$\langle 3, 2 \rangle$	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	$\langle 4, 3 \rangle$	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	$\langle 6, 4 \rangle$	0	0	3	4	6	7	9	10	10	13	13	13	13	13	13	13
4	$\langle 5, 5 \rangle$	0	0	3	4	6											
5	$\langle 5, 6 \rangle$	0															

Diagram illustrating the knapsack problem with 5 objects and a capacity of 15. The table shows the maximum value for each capacity (0 to 15) considering the objects up to the current row.

Annotations:

- Red arrow from (4, 6) to (3, 7): Salto: -1
- Red arrow from (4, 6) to (5, 0): Salto: -5 (w_4)

El problema de la mochila

Objeto		Capacidad de la mochila															
	$\langle v, w \rangle$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	$\langle 3, 2 \rangle$	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	$\langle 4, 3 \rangle$	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	$\langle 6, 4 \rangle$	0	0	3	4	6	7	9	10	10	13	13	13	13	13	13	13
4	$\langle 5, 5 \rangle$	0	0	3	4	6	max										
5	$\langle 5, 6 \rangle$	0															

El problema de la mochila

Objeto

Capacidad de la mochila

	<v, w>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	<3, 2>	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	<4, 3>	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	<6, 4>	0	0	3	4	6	7	9	10	10	13	13	13	13	13	13	13
4	<5, 5>	0	0	3	4	6	7										
5	<5, 6>	0															

Diagram illustrating the knapsack problem. The table shows the maximum value (v) that can be achieved for each capacity (w) from 0 to 15, considering items 1 through 5. The values are calculated based on the items' weights (w) and values (v).

Key annotations:

- A red arrow points from the value 0 in the row for item 3, capacity 0, to the value 7 in the row for item 4, capacity 5.
- A red arrow points from the value 7 in the row for item 3, capacity 5, to the value 7 in the row for item 4, capacity 5.
- The value 7 in the row for item 4, capacity 5, is highlighted with a black box.
- The label $+v_4$ is placed below the cell (4, 5), indicating the addition of item 4's value to the previous state.

El problema de la mochila

[illegible]

El problema de la mochila

[illegible]

El problema de la mochila

[illegible]

El problema de la mochila

[illegible]

El problema de la mochila

Objeto

Capacidad de la mochila

	<v, w>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	<3, 2>	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	<4, 3>	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	<6, 4>	0	0	3	4	6	7	9	10	10	13	13	13	13	13	13	13
4	<5, 5>	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18
5	<5, 6>	0	0	3	4	6	7										

Salto: -6 (w_5)

Salto: -1

El problema de la mochila

Objeto

Capacidad de la mochila

	<v, w>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	<3, 2>	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	<4, 3>	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	<6, 4>	0	0	3	4	6	7	9	10	10	13	13	13	13	13	13	13
4	<5, 5>	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18
5	<5, 6>	0	0	3	4	6	7	max									

$+v_5$

El problema de la mochila

Objeto

Capacidad de la mochila

	<v, w>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	<3, 2>	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	<4, 3>	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	<6, 4>	0	0	3	4	6	7	9	10	10	13	13	13	13	13	13	13
4	<5, 5>	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18
5	<5, 6>	0	0	3	4	6	7	9									

$+v_5$

El problema de la mochila

[illegible]

El problema de la mochila

Objeto		Capacidad de la mochila															
	$\langle v, w \rangle$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	$\langle 3, 2 \rangle$	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	$\langle 4, 3 \rangle$	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	$\langle 6, 4 \rangle$	0	0	3	4	6	7	9	10	10	13	13	13	13	13	13	13
4	$\langle 5, 5 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18
5	$\langle 5, 6 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	

El problema de la mochila

Objeto		Capacidad de la mochila															
↓	<v, w>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	1 <3, 2>	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
	2 <4, 3>	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
	3 <6, 4>	0	0	3	4	6	7	9	10	10	13	13	13	13	13	13	13
	4 <5, 5>	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18
	5 <5, 6>	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	

El problema de la mochila

Objeto

Capacidad de la mochila

	<v, w>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	<3, 2>	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	<4, 3>	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	<6, 4>	0	0	3	4	6	7	9	10	10	13	13	13	13	13	13	13
4	<5, 5>	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18
5	<5, 6>	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	

Salto: -6 (w_5)

Salto: -1

El problema de la mochila

Objeto

Capacidad de la mochila

		Capacidad de la mochila															
Objeto ↓	<v, w>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	1 <3, 2>	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
	2 <4, 3>	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
	3 <6, 4>	0	0	3	4	6	7	9	10	10	13	13	13	13	13	13	13
	4 <5, 5>	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18
	5 <5, 6>	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	max

$+v_5$


Se produce
un empate
→ dos
posibles
soluciones

El problema de la mochila

Objeto

Capacidad de la mochila

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	<3, 2>	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	<4, 3>	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	<6, 4>	0	0	3	4	6	7	9	10	10	13	13	13	13	13	13	13
4	<5, 5>	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18
5	<5, 6>	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18



$+v_5$

El problema de la mochila

Objeto

Capacidad de la mochila

	<v, w>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	<3, 2>	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	<4, 3>	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	<6, 4>	0	0	3	4	6	7	9	10	10	13	13	13	13	13	13	13
4	<5, 5>	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18
5	<5, 6>	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18

Máximo valor de la carga para una mochila de W=15

El problema de la mochila

Objeto		Capacidad de la mochila															
	$\langle v, w \rangle$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	$\langle 3, 2 \rangle$	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	$\langle 4, 3 \rangle$	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	$\langle 6, 4 \rangle$	0	0	3	4	6	7	9	10	10	13	13	13	13	13	13	13
4	$\langle 5, 5 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18
5	$\langle 5, 6 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18

¿Cuáles son los objetos?:
Construir la traza desde este
elemento $(v[n, W])$ hasta $v[0, 0]$.

En este caso tenemos dos
posibles soluciones.

El problema de la mochila


Objeto		Capacidad de la mochila															
	$\langle v, w \rangle$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	$\langle 3, 2 \rangle$	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	$\langle 4, 3 \rangle$	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	$\langle 6, 4 \rangle$	0	0	3	4	6	7	9	10	10	13	13	13	13	13	13	13
4	$\langle 5, 5 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18
5	$\langle 5, 6 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18

Solución 1

El problema de la mochila

Objeto

Capacidad de la mochila

		Capacidad de la mochila															
	<v, w>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	1 <3, 2>	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
	2 <4, 3>	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
	3 <6, 4>	0	0	3	4	6	7	9	10	10	13	13	13	13	13	13	13
	4 <5, 5>	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18
	5 <5, 6>	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18

No usar el
objeto 5

Solución 1:

El problema de la mochila

Objeto		Capacidad de la mochila															
	$\langle v, w \rangle$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	$\langle 3, 2 \rangle$	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	$\langle 4, 3 \rangle$	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	$\langle 6, 4 \rangle$	0	0	3	4	6	7	9	10	10	13	13	13	13	13	13	13
4	$\langle 5, 5 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18
5	$\langle 5, 6 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18

Solución 1:

El problema de la mochila

Objeto		Capacidad de la mochila															
	$\langle v, w \rangle$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	$\langle 3, 2 \rangle$	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	$\langle 4, 3 \rangle$	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	$\langle 6, 4 \rangle$	0	0	3	4	6	7	9	10	10	13	13	13	13	13	13	13
4	$\langle 5, 5 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18
5	$\langle 5, 6 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18

Solución 1:



Objeto 4

El problema de la mochila

Objeto		Capacidad de la mochila															
	$\langle v, w \rangle$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	$\langle 3, 2 \rangle$	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	$\langle 4, 3 \rangle$	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	$\langle 6, 4 \rangle$	0	0	3	4	6	7	9	10	10	13	13	13	13	13	13	13
4	$\langle 5, 5 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18
5	$\langle 5, 6 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18

Salto: -5 (w_4)

Solución 1:



Objeto 4

El problema de la mochila

Objeto		Capacidad de la mochila															
	$\langle v, w \rangle$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	$\langle 3, 2 \rangle$	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	$\langle 4, 3 \rangle$	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	$\langle 6, 4 \rangle$	0	0	3	4	6	7	9	10	10	13	13	13	13	13	13	13
4	$\langle 5, 5 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18
5	$\langle 5, 6 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18

Solución 1:



Objeto 4

El problema de la mochila

Objeto		Capacidad de la mochila															
	$\langle v, w \rangle$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	$\langle 3, 2 \rangle$	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	$\langle 4, 3 \rangle$	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	$\langle 6, 4 \rangle$	0	0	3	4	6	7	9	10	10	13	13	13	13	13	13	13
4	$\langle 5, 5 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18
5	$\langle 5, 6 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18

Solución 1:



Objeto 3



Objeto 4

El problema de la mochila

Objeto		Capacidad de la mochila															
	$\langle v, w \rangle$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	$\langle 3, 2 \rangle$	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	$\langle 4, 3 \rangle$	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	$\langle 6, 4 \rangle$	0	0	3	4	6	7	9	10	10	13	13	13	13	13	13	13
4	$\langle 5, 5 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18
5	$\langle 5, 6 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18

Salto: -4 (w_3)

Solución 1:



Objeto 3



Objeto 4

El problema de la mochila

Objeto		Capacidad de la mochila															
	$\langle v, w \rangle$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	$\langle 3, 2 \rangle$	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	$\langle 4, 3 \rangle$	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	$\langle 6, 4 \rangle$	0	0	3	4	6	7	9	10	10	13	13	13	13	13	13	13
4	$\langle 5, 5 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18
5	$\langle 5, 6 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18

Solución 1:



Objeto 3



Objeto 4

El problema de la mochila

Objeto		Capacidad de la mochila															
	$\langle v, w \rangle$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	$\langle 3, 2 \rangle$	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	$\langle 4, 3 \rangle$	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	$\langle 6, 4 \rangle$	0	0	3	4	6	7	9	10	10	13	13	13	13	13	13	13
4	$\langle 5, 5 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18
5	$\langle 5, 6 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18

Solución 1:



Objeto 2



Objeto 3



Objeto 4

El problema de la mochila

Objeto		Capacidad de la mochila															
	$\langle v, w \rangle$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	$\langle 3, 2 \rangle$	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	$\langle 4, 3 \rangle$	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	$\langle 6, 4 \rangle$	0	0	3	4	6	7	9	10	10	13	13	13	13	13	13	13
4	$\langle 5, 5 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18
5	$\langle 5, 6 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18

Salto: -3 (w_2)

Solución 1:



Objeto 2



Objeto 3



Objeto 4

El problema de la mochila

Objeto		Capacidad de la mochila															
	$\langle v, w \rangle$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	$\langle 3, 2 \rangle$	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	$\langle 4, 3 \rangle$	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	$\langle 6, 4 \rangle$	0	0	3	4	6	7	9	10	10	13	13	13	13	13	13	13
4	$\langle 5, 5 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18
5	$\langle 5, 6 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18

Solución 1:



Objeto 2



Objeto 3



Objeto 4

El problema de la mochila

Objeto		Capacidad de la mochila															
	$\langle v, w \rangle$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	$\langle 3, 2 \rangle$	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	$\langle 4, 3 \rangle$	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	$\langle 6, 4 \rangle$	0	0	3	4	6	7	9	10	10	13	13	13	13	13	13	13
4	$\langle 5, 5 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18
5	$\langle 5, 6 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18

Solución 1:

$V = 18$

$W = 14$



Objeto 1



Objeto 2



Objeto 3



Objeto 4

El problema de la mochila

Objeto		Capacidad de la mochila															
↓	<v, w>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	<3, 2>	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	<4, 3>	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	<6, 4>	0	0	3	4	6	7	9	10	10	13	13	13	13	13	13	13
4	<5, 5>	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18
5	<5, 6>	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18

Solución 2

El problema de la mochila

Objeto		Capacidad de la mochila															
	$\langle v, w \rangle$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	$\langle 3, 2 \rangle$	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	$\langle 4, 3 \rangle$	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	$\langle 6, 4 \rangle$	0	0	3	4	6	7	9	10	10	13	13	13	13	13	13	13
4	$\langle 5, 5 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18
5	$\langle 5, 6 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18

Solución 2:



Objeto 5

El problema de la mochila

Objeto		Capacidad de la mochila															
	$\langle v, w \rangle$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	$\langle 3, 2 \rangle$	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	$\langle 4, 3 \rangle$	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	$\langle 6, 4 \rangle$	0	0	3	4	6	7	9	10	10	13	13	13	13	13	13	13
4	$\langle 5, 5 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18
5	$\langle 5, 6 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18

Salto: -6 (w_5)

Solución 2:



Objeto 5

El problema de la mochila

Objeto		Capacidad de la mochila															
	$\langle v, w \rangle$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	$\langle 3, 2 \rangle$	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	$\langle 4, 3 \rangle$	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	$\langle 6, 4 \rangle$	0	0	3	4	6	7	9	10	10	13	13	13	13	13	13	13
4	$\langle 5, 5 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18
5	$\langle 5, 6 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18

Solución 2:



Objeto 5

El problema de la mochila

Objeto		Capacidad de la mochila															
	$\langle v, w \rangle$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	$\langle 3, 2 \rangle$	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	$\langle 4, 3 \rangle$	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	$\langle 6, 4 \rangle$	0	0	3	4	6	7	9	10	10	13	13	13	13	13	13	13
4	$\langle 5, 5 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18
5	$\langle 5, 6 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18

No usar el objeto 4

Solución 2:



Objeto 5

El problema de la mochila

Objeto		Capacidad de la mochila															
	$\langle v, w \rangle$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	$\langle 3, 2 \rangle$	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	$\langle 4, 3 \rangle$	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	$\langle 6, 4 \rangle$	0	0	3	4	6	7	9	10	10	13	13	13	13	13	13	13
4	$\langle 5, 5 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18
5	$\langle 5, 6 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18

Solución 2:



Objeto 5

El problema de la mochila

Objeto		Capacidad de la mochila															
	$\langle v, w \rangle$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	$\langle 3, 2 \rangle$	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	$\langle 4, 3 \rangle$	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	$\langle 6, 4 \rangle$	0	0	3	4	6	7	9	10	10	13	13	13	13	13	13	13
4	$\langle 5, 5 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18
5	$\langle 5, 6 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18

Solución 2:



Objeto 3



Objeto 5

El problema de la mochila

Objeto		Capacidad de la mochila															
	$\langle v, w \rangle$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	$\langle 3, 2 \rangle$	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	$\langle 4, 3 \rangle$	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	$\langle 6, 4 \rangle$	0	0	3	4	6	7	9	10	10	13	13	13	13	13	13	13
4	$\langle 5, 5 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18
5	$\langle 5, 6 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18

Salto: -4 (w_3)

Solución 2:



Objeto 3



Objeto 5

El problema de la mochila

Objeto		Capacidad de la mochila															
	$\langle v, w \rangle$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	$\langle 3, 2 \rangle$	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	$\langle 4, 3 \rangle$	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	$\langle 6, 4 \rangle$	0	0	3	4	6	7	9	10	10	13	13	13	13	13	13	13
4	$\langle 5, 5 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18
5	$\langle 5, 6 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18

Solución 2:



Objeto 3



Objeto 5

El problema de la mochila

Objeto		Capacidad de la mochila															
	$\langle v, w \rangle$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	$\langle 3, 2 \rangle$	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	$\langle 4, 3 \rangle$	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	$\langle 6, 4 \rangle$	0	0	3	4	6	7	9	10	10	13	13	13	13	13	13	13
4	$\langle 5, 5 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18
5	$\langle 5, 6 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18

Solución 2:



Objeto 2



Objeto 3



Objeto 5

El problema de la mochila

Objeto

Capacidad de la mochila

	<v, w>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	<3, 2>	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	<4, 3>	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	<6, 4>	0	0	3	4	6	7	9	10	10	13	13	13	13	13	13	13
4	<5, 5>	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18
5	<5, 6>	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18

Salto: -3 (w_2)

Solución 2:



Objeto 2



Objeto 3



Objeto 5

El problema de la mochila

Objeto		Capacidad de la mochila															
	$\langle v, w \rangle$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	$\langle 3, 2 \rangle$	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	$\langle 4, 3 \rangle$	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	$\langle 6, 4 \rangle$	0	0	3	4	6	7	9	10	10	13	13	13	13	13	13	13
4	$\langle 5, 5 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18
5	$\langle 5, 6 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18

Solución 2:



Objeto 2



Objeto 3



Objeto 5

El problema de la mochila

Objeto		Capacidad de la mochila															
	$\langle v, w \rangle$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	$\langle 3, 2 \rangle$	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	$\langle 4, 3 \rangle$	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	$\langle 6, 4 \rangle$	0	0	3	4	6	7	9	10	10	13	13	13	13	13	13	13
4	$\langle 5, 5 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18
5	$\langle 5, 6 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18

Solución 2:

V = 18
W = 15



Objeto 1



Objeto 2



Objeto 3



Objeto 5

El problema de la mochila

Objeto		Capacidad de la mochila															
	$\langle v, w \rangle$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	$\langle 3, 2 \rangle$	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	$\langle 4, 3 \rangle$	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	$\langle 6, 4 \rangle$	0	0	3	4	6	7	9	10	10	13	13	13	13	13	13	13
4	$\langle 5, 5 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18
5	$\langle 5, 6 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18

¿Qué solución proporcionaría el algoritmo voraz en este caso?

El problema de la mochila

Objeto		Capacidad de la mochila															
	$\langle v, w \rangle$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	$\langle 3, 2 \rangle$	0	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	$\langle 4, 3 \rangle$	0	0	3	4	4	7	7	7	7	7	7	7	7	7	7	7
3	$\langle 6, 4 \rangle$	0	0	3	4	6	7	9	10	10	13	13	13	13	13	13	13
4	$\langle 5, 5 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18
5	$\langle 5, 6 \rangle$	0	0	3	4	6	7	9	10	10	13	13	14	15	15	18	18

Solución del algoritmo voraz:

Se calculan rentabilidades, en este caso $r = [1,50; 1,33; 1,50; 1,00; 0,83]$.

Al elegir los objetos por orden de rentabilidad obtendríamos 1, 3, 2, 4.

Coincide con la primera solución del algoritmo de programación dinámica, pero únicamente por casualidad...

El problema de la mochila: parte II

Ejercicio 3: diseñar el pseudocódigo de una función *Composición* que a partir de la tabla M , ya construida en el ejercicio anterior, devuelva una configuración posible para la carga óptima, especificando el conjunto de objetos que la componen. Proponer un tipo de datos adecuado para la salida de la función *Composición*. Determinar igualmente la complejidad de esta función.

El problema de la mochila: parte II

- Estructura a devolver en la función *Composición*:
 - Vector de valores booleanos: $C[1..n]$
- Para la solución 2 del ejercicio anterior se obtendría: [V, V, V, F, V]

El problema de la mochila: parte II

```
función Composición ( M[1..n,0..W], w[1..n] ) : C[1..n]
  para i:=1 hasta n hacer C[i]:=false;
  v:=M[n,W];
  i:=n; j:=W;
  mientras i>1 y v>0 hacer
    si M[i,j] <> M[i-1,j] entonces % el objeto i está en la configuración
      C[i]:=true;
      j:=j-w[i];
      v:=M[i-1,j] % leer el valor de la tabla evita calcular v-v[i]
    fin si;
    i:=i-1
  fin mientras;

  si v>0 entonces C[1]:=true; % caso particular: i=1

  devolver C
fin función
```


El problema de la mochila: parte II

```
función Composición ( M[1..n,0..W], w[1..n] ) : C[1..n]
  para i:=1 hasta n hacer C[i]:=false;  $\{\theta(n)\}$ 
  v:=M[n,W];
  i:=n; j:=W;
  mientras i>1 y v>0 hacer
    si M[i,j] <> M[i-1,j] entonces % el objeto i está en la configuración
      C[i]:=true;
      j:=j-w[i];
      v:=M[i-1,j] % leer el valor de la tabla evita calcular v-v[i]
    fin si;
    i:=i-1
  fin mientras;
  si v>0 entonces C[1]:=true; % caso particular: i=1
  devolver C
fin función
```

El problema de la mochila: parte II

función Composición ($M[1..n, 0..W]$, $w[1..n]$) : $C[1..n]$

para $i:=1$ **hasta** n **hacer** $C[i]:=false$; $\{\theta(n)\}$

$v:=M[n,W]$; $\{\theta(1)\}$

$i:=n$; $j:=W$; $\{\theta(1)\}$

mientras $i>1$ **y** $v>0$ **hacer**

si $M[i,j] \neq M[i-1,j]$ **entonces** % el objeto i está en la configuración

$C[i]:=true$;

$j:=j-w[i]$;

$v:=M[i-1,j]$ % leer el valor de la tabla evita calcular $v-v[i]$

fin si;

$i:=i-1$

fin mientras;

si $v>0$ **entonces** $C[1]:=true$; % caso particular: $i=1$

devolver C

fin función

El problema de la mochila: parte II

función Composición ($M[1..n, 0..W]$, $w[1..n]$) : $C[1..n]$

para $i:=1$ **hasta** n **hacer** $C[i]:=false$; $\{\theta(n)\}$

$v:=M[n,W]$; $\{\theta(1)\}$

$i:=n$; $j:=W$; $\{\theta(1)\}$

mientras $i>1$ **y** $v>0$ **hacer**

$\{\theta(1)\}$ **si** $M[i,j] \neq M[i-1,j]$ **entonces** % el objeto i está en la configuración

$\{\theta(1)\}$ $C[i]:=true$;

$\{\theta(1)\}$ $j:=j-w[i]$;

$\{\theta(1)\}$ $v:=M[i-1,j]$ % leer el valor de la tabla evita calcular $v-v[i]$

fin si;

$i:=i-1$ $\{\theta(1)\}$

fin mientras;

si $v>0$ **entonces** $C[1]:=true$; % caso particular: $i=1$

devolver C

fin función

El problema de la mochila: parte II

función Composición ($M[1..n, 0..W]$, $w[1..n]$) : $C[1..n]$

para $i:=1$ **hasta** n **hacer** $C[i]:=false$; $\{\theta(n)\}$

$v:=M[n,W]$; $\{\theta(1)\}$

$i:=n$; $j:=W$; $\{\theta(1)\}$

mientras $i>1$ **y** $v>0$ **hacer** $\{O(n)\}$

$\{\theta(1)\}$ **si** $M[i,j] \neq M[i-1,j]$ **entonces** % el objeto i está en la configuración

$\{\theta(1)\}$ $C[i]:=true$;

$\{\theta(1)\}$ $j:=j-w[i]$;

$\{\theta(1)\}$ $v:=M[i-1,j]$ % leer el valor de la tabla evita calcular $v-v[i]$

fin si;

$i:=i-1$ $\{\theta(1)\}$

fin mientras;

si $v>0$ **entonces** $C[1]:=true$; % caso particular: $i=1$

devolver C

fin función

El problema de la mochila: parte II

función Composición ($M[1..n, 0..W]$, $w[1..n]$) : $C[1..n]$

para $i:=1$ **hasta** n **hacer** $C[i]:=false$; $\{\theta(n)\}$

$v:=M[n,W]$; $\{\theta(1)\}$

$i:=n$; $j:=W$; $\{\theta(1)\}$

mientras $i>1$ **y** $v>0$ **hacer** $\{O(n)\}$

$\{\theta(1)\}$ **si** $M[i,j] \neq M[i-1,j]$ **entonces** % el objeto i está en la configuración

$\{\theta(1)\}$ $C[i]:=true$;

$\{\theta(1)\}$ $j:=j-w[i]$;

$\{\theta(1)\}$ $v:=M[i-1,j]$ % leer el valor de la tabla evita calcular $v-v[i]$

fin si;

$i:=i-1$ $\{\theta(1)\}$

fin mientras;

si $v>0$ **entonces** $C[1]:=true$; % caso particular: $i=1$ $\{\theta(1)\}$

devolver C $\{\theta(1)\}$

fin función

El problema de la mochila: parte II

función Composición ($M[1..n, 0..W]$, $w[1..n]$) : $C[1..n]$ $\{\theta(n)\}$

para $i:=1$ **hasta** n **hacer** $C[i]:=false$; $\{\theta(n)\}$

$v:=M[n,W]$; $\{\theta(1)\}$

$i:=n$; $j:=W$; $\{\theta(1)\}$

mientras $i>1$ **y** $v>0$ **hacer** $\{O(n)\}$

$\{\theta(1)\}$ **si** $M[i,j] \neq M[i-1,j]$ **entonces** % el objeto i está en la configuración

$\{\theta(1)\}$ $C[i]:=true$;

$\{\theta(1)\}$ $j:=j-w[i]$;

$\{\theta(1)\}$ $v:=M[i-1,j]$ % leer el valor de la tabla evita calcular $v-v[i]$

fin si;

$i:=i-1$ $\{\theta(1)\}$

fin mientras;

si $v>0$ **entonces** $C[1]:=true$; % caso particular: $i=1$ $\{\theta(1)\}$

devolver C $\{\theta(1)\}$

fin función