



Seminario: Programación dinámica

José María Casanova, Elena Hernández, Alberto Valderruten,
Óscar Fontenla Romero y Carlos Eiras

Dept. Ciencias de la Computación y Tecnologías de la Información

`{jose.casanova.crespo, elena.hernandez, alberto.valderruten,
oscar.fontenla, carlos.eiras.franco}@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

[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

The diagram shows a table for the coin change problem. The columns represent the amount to be paid (0 to 17), and the rows represent the coin denominations (0, 1, 3, 8, 12). The top row (0) is the header. The first column (0) is the header. The table contains the following values:

	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																	

Red arrows and text indicate jumps:

- A red arrow points from the cell at (3, 4) to the cell at (3, 3), labeled **Salto: -1**.
- A red arrow points from the cell at (3, 4) to the cell at (8, 1), labeled **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 memory layout with indices 0 to 17. A vertical arrow on the left indicates increasing index values (1, 3, 8, 12). A horizontal bracket at the top indicates a range of indices (0 to 17). A red arrow points from index 5 to index 3, labeled "Salto: -1". Another red arrow points from index 5 to index 2, labeled "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

[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	1									
12	0																	

Salto: -1

Salto: -8

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	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: -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					

Devolver el cambio

Denominación
de la moneda

Importe a pagar

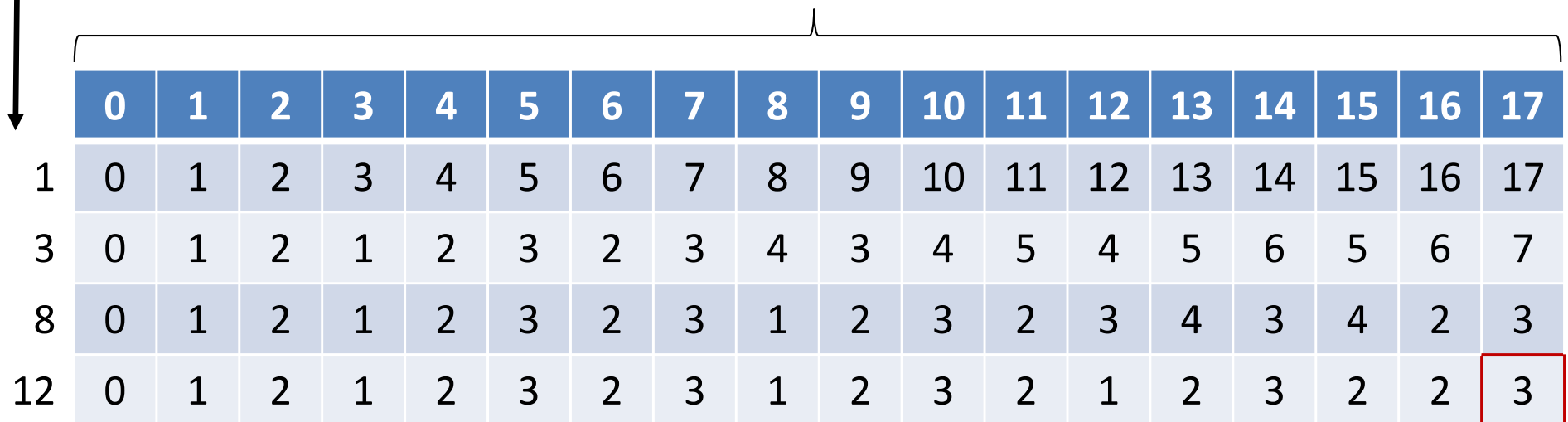


	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
3	0	1	2	1	2	3	2	3	4	3	4	5	4	5	6	5	6	7
8	0	1	2	1	2	3	2	3	1	2	3	2	3	4	3	4	2	3
12	0	1	2	1	2	3	2	3	1	2	3	2	1	2	3	2	2	3

Devolver el cambio

Denominación
de la moneda

Importe a pagar



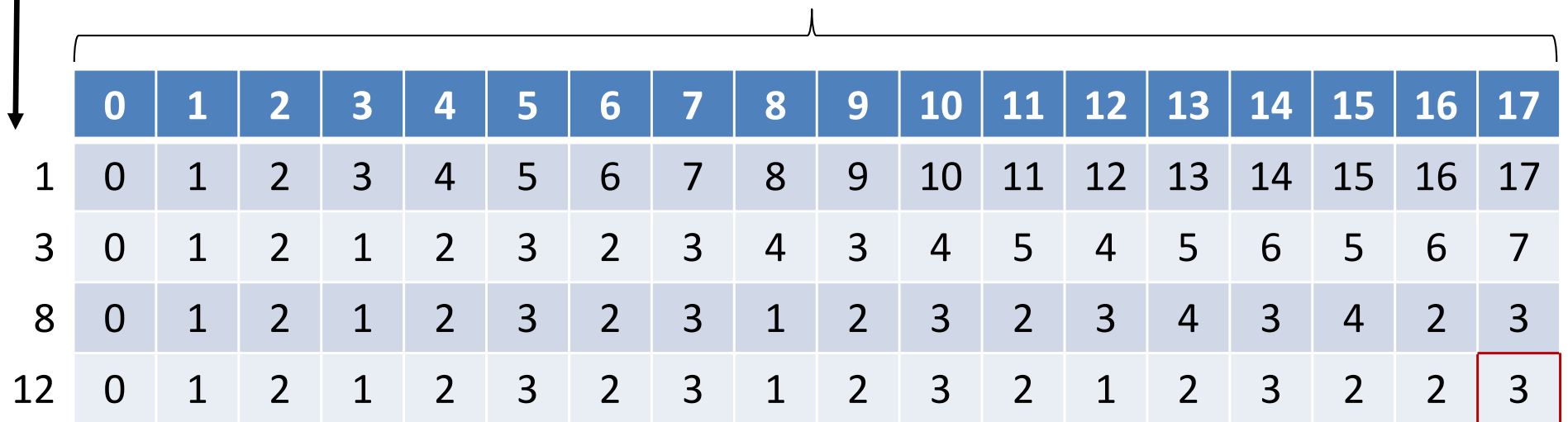
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
3	0	1	2	1	2	3	2	3	4	3	4	5	4	5	6	5	6	7
8	0	1	2	1	2	3	2	3	1	2	3	2	3	4	3	4	2	3
12	0	1	2	1	2	3	2	3	1	2	3	2	1	2	3	2	2	3

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

	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

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



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

	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

		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

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

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 with a table of values and annotations:

- Row 2, Column 4 (value 3) is highlighted with a red box.
- Red arrows indicate jumps from the highlighted cell to Row 1, Column 3 (value 3) and Row 3, Column 1 (value 0).
- Annotations: "Salto: -1" (from Row 2, Column 4 to Row 1, Column 3) and "Salto: -3 (w_2)" (from Row 2, Column 4 to Row 3, Column 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	max												
3	<6, 4>	0		+v ₂													
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												
3	<6, 4>	0															
4	<5, 5>	0															
5	<5, 6>	0															

Diagram illustrating the knapsack problem. The table shows the maximum value for items 1 to 5 across capacities 0 to 15. The value for item 2 at capacity 4 is highlighted as 4, which is the sum of the value of item 1 (3) and the value of item 2 (1) at capacity 1 (0). The red arrow indicates the calculation: $0 + v_2 = 4$.

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												
3	$\langle 6, 4 \rangle$	0															
4	$\langle 5, 5 \rangle$	0															
5	$\langle 5, 6 \rangle$	0															

Diagram illustrating the knapsack problem with a table of values and annotations:

- The table shows the maximum value for each capacity (0 to 15) and each object (1 to 5).
- Object 1: $\langle 3, 2 \rangle$ (Value 3, Weight 2). The value is 3 for capacities 3 through 15.
- Object 2: $\langle 4, 3 \rangle$ (Value 4, Weight 3). The value is 4 for capacity 4. A red arrow points from the value 4 to the value 3 at capacity 5, labeled "Salto: -1".
- Object 3: $\langle 6, 4 \rangle$ (Value 6, Weight 4). The value is 6 for capacity 4. A red arrow points from the value 6 to the value 3 at capacity 5, labeled "Salto: -3 (w_2)".

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															

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											
3	$\langle 6, 4 \rangle$	0															
4	$\langle 5, 5 \rangle$	0															
5	$\langle 5, 6 \rangle$	0															

Diagram illustrating the knapsack problem with a table of values and annotations:

- The table shows the maximum value v for each capacity w (0 to 15) for each object (1 to 5).
- Object 1: $\langle 3, 2 \rangle$. Values are 0 for $w < 3$ and 3 for $w \geq 3$.
- Object 2: $\langle 4, 3 \rangle$. Values are 0 for $w < 4$, 3 for $w = 4$, and 4 for $w = 5$. The cell at $(2, 5)$ is empty.
- Object 3: $\langle 6, 4 \rangle$. Values are 0 for $w < 6$ and empty for $w \geq 6$.
- Object 4: $\langle 5, 5 \rangle$. Values are 0 for $w < 5$ and empty for $w \geq 5$.
- Object 5: $\langle 5, 6 \rangle$. Values are 0 for $w < 5$ and empty for $w \geq 5$.

Annotations:

- A red arrow points from the cell at $(2, 5)$ to the cell at $(2, 4)$, labeled "Salto: -1".
- A red arrow points from the cell at $(2, 4)$ to the cell at $(2, 3)$, labeled "Salto: -3 (w_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	max										
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 for each capacity (0 to 15) and each object (1 to 5). The values are calculated based on the object's value (v) and weight (w).

Key annotations:

- Red arrows indicate the calculation of the maximum value for capacity 5, showing the transition from capacity 2 (value 3) to capacity 5 (value 4) using object 2.
- The cell for capacity 5 and object 2 is labeled "max", indicating the optimal solution for that capacity.
- The label $+v_2$ indicates the addition of the value of object 2 to the previous maximum value.

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

[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

	<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	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											
4	<5, 5>	0															
5	<5, 6>	0															

Diagram illustrating the knapsack problem. The table shows the maximum value for items 1 to 5 across capacities 0 to 15. Red arrows and text highlight the calculation for item 3 at capacity 6: a curved arrow from (2, 3) to (3, 4) is labeled $+v_3$, and a straight arrow from (2, 4) to (3, 6) is labeled 4.

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											
4	$\langle 5, 5 \rangle$	0															
5	$\langle 5, 6 \rangle$	0															

Diagram illustrating the knapsack problem with a table of values and annotations:

- The table shows the maximum value for each capacity (0 to 15) and each object (1 to 5).
- Object 1: $\langle 3, 2 \rangle$ (Value 3, Weight 2)
- Object 2: $\langle 4, 3 \rangle$ (Value 4, Weight 3)
- Object 3: $\langle 6, 4 \rangle$ (Value 6, Weight 4)
- Object 4: $\langle 5, 5 \rangle$ (Value 5, Weight 5)
- Object 5: $\langle 5, 6 \rangle$ (Value 5, Weight 6)

Annotations:

- A red arrow points from the cell (3, 4) to the cell (4, 6), labeled "Salto: -4 (w_3)".
- A red arrow points from the cell (4, 6) to the cell (5, 7), labeled "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	max										
4	<5, 5>	0															
5	<5, 6>	0															

Diagram illustrating the knapsack problem. The table shows the maximum value for different items and capacities. Red arrows highlight the calculation of the maximum value for item 3 at capacity 6, showing it is the maximum of the previous value (6) and the value of item 2 plus its weight (4 + 3 = 7).

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

	<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											
5	<5, 6>	0															

Salto: -1

Salto: -5 (w_4)

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	max										
5	<5, 6>	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	7	9	10	10	13	13	13	13	13	13	13
4	$\langle 5, 5 \rangle$	0	0	3	4	6	7										
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 items 1 through 5. The items are defined by their value v and weight w as $\langle v, w \rangle$.

Red arrows and annotations highlight the calculation for item 4 at capacity 7:

- A red arrow points from the value 0 at $(w=0, \text{item}=4)$ to the value 7 at $(w=7, \text{item}=4)$.
- A red arrow points from the value 7 at $(w=6, \text{item}=3)$ to the value 7 at $(w=7, \text{item}=4)$.
- The annotation $+v_4$ is placed below the cell at $(w=4, \text{item}=5)$, indicating the value added by item 4.

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															
	<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	

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

	<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

+v₅

Se produce un empate → dos posibles soluciones

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

$+v_5$

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

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															
	$\langle v, w \rangle$	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 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															
	$\langle v, w \rangle$	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

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:



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															
↓	<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:



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															
	$\langle v, w \rangle$	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 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] <> 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] <> 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] <> 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] <> 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] <> 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
```