

MONEDAS INFINITAS

TENEMOS N MONEDAS $x_1 \dots x_n$ ORDENADAS MENOR A MAYOR

HAY QUE HACERLO POR h

$$NM = \min_{0 \leq k \leq \left\lfloor \frac{c}{v_i} \right\rfloor} \left(\left\lfloor \frac{c}{v_i} \right\rfloor - k \right) + NM(i-1, c - v_i * k)$$

NO PUEDE SER NEGATIVO

NUMERO DE MONEDAS DE ESE TIPO

q_i

	8	
--	---	--

$i=0 \wedge c \bmod v_0 \neq 0 \parallel \frac{c}{v_0} > q_0 \infty$

$$i=0 \wedge \frac{c}{v_0} \overline{A \cup B} \quad \overline{A \cap B}$$

TENEMOS QUE BUSCAR EL MINIMO DE MONEDAS

PARA ENTENDERLO MEJOR CON LAS TABLAS

	0	1	2	3	4	5	6	7	8	M
0	0	∞	∞	∞	∞	∞	∞	∞	∞	
2	0	∞	1	∞	2	∞	3			
3	0	∞	1	1	2	2	2			
7	0	∞	1	1	2	2	2			

$$\text{cambio}(i, j) = \begin{cases} \text{cambio}(i-1, j) & \text{si } X_i > j \\ \min \{ \text{cambio}(i-1, j), \text{cambio}(i, j - X_k) + 1 \} & \text{si } X_i \leq j \end{cases}$$



VE:1

	0	1	2	3	4	5	6	7	8	M
0	0	∞	∞	∞	∞	∞	∞	∞	∞	
1	2	0	∞	1	∞	2	∞	3	∞	4
2	3	0	∞	1	1	2	2	2	3	3
3	7	0	∞	1	1	2	2	2	1	3

si $(i > 1 \wedge t[i, j] = t[i-1, j]) \rightarrow i := i - 1$
 sino $\rightarrow M[i]++; j := j - X_i$

	0	1	2	3	4	5	6	7	8	M
0	0	∞	∞	∞	∞	∞	∞	∞	∞	
2	0	∞	1	∞	2	∞	3	∞	4	
3	0	∞	1	1	2	2	2	3	3	2
7	0	∞	1	1	2	2	2	1	3	

$si(i > 1 \wedge t[i, j] = t[i-1, j]) \rightarrow i := i - 1$
 $sino \rightarrow M[i]++; j := j - X_i$

	0	1	2	3	4	5	6	7	8	M
0	0	∞	∞	∞	∞	∞	∞	∞	∞	
2	0	∞	1	∞	2	∞	3	∞	4	1
3	0	∞	1	1	2	2	2	3	3	2
7	0	∞	1	1	2	2	2	1	3	

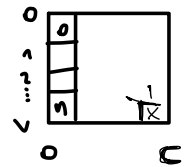
Handwritten annotations on the table:

- Row 0, Column 0: circled in purple with "11" below it.
- Row 2, Column 2: circled in red with "11" below it.
- Row 2, Column 8: circled in green with "1" to its right.
- Row 3, Column 8: circled in blue with "2" to its right.
- Row 7, Column 8: circled in blue with "11" below it.
- Purple arrows: from (0,0) to (2,2) and (2,8).
- Red arrows: from (2,2) to (3,3), (3,4), (3,5), (3,6), (3,7), and (3,8).
- Red "X" marks: above (2,2), (3,5), and (3,8).
- Red "+1" labels: above (2,8), (3,6), and (3,8).
- Green "+1" label: to the right of (2,8).
- Red "1+1" label: to the right of (3,8).

v - CANTIDAD DE MONEDAS DISTINTAS

C - CANTIDAD A LLEGAR

q - NUMERO DE MONEDAS DE ESE TIPO



```
public static Res cambio(int[] v, int[] q, int C) {
    int n = v.length; // Obtiene la cantidad de tipos de monedas disponibles.
    int[][] NM = new int[n][C + 1]; // Crea una matriz para almacenar las soluciones.

    for (int i = 0; i < n; i++) { // Recorre los diferentes tipos de monedas disponibles.
        for (int c = 0; c <= C; c++) { // Recorre las cantidades desde 0 hasta C.

            if (i == 0 && (c % v[0] != 0 || c / v[0] > q[0])) {  $\rightarrow$  usar  $\infty$ 
                // Verifica si estamos en el primer tipo de moneda y si la cantidad no es un múltiplo
                // de la denominación de la moneda o si supera la cantidad máxima permitida.
                NM[i][c] = Integer.MAX_VALUE / 2; // Establece un valor alto para evitar usar esta opción.
            } else if (i == 0) {
                NM[i][c] = c / v[0]; // Calcula la cantidad de monedas del primer tipo que se pueden usar.
            } else {
                NM[i][c] = NM[i - 1][c]; // Inicializa con la opción de no usar el tipo de moneda actual.

                for (int k = 1; k <= Math.min(q[i], c / v[i]); k++) {
                    // Comprueba cuántas monedas del tipo actual se pueden usar.
                    NM[i][c] = Math.min(NM[i][c], NM[i - 1][c - v[i] * k] + k);
                    // Calcula el mínimo entre no usar el tipo actual y usarlo k veces.
                }
            }
        }
    }
}
```