Programación dinámica, problema de cambio con varios billetes de cada tipo (limitados).
Autor: Luis Quesada Torres.

Este es el problema de tipo más complejo de todos los que se pueden hacer de programación dinámica, por ello incluyo un ejemplo completo.

Enunciado:

Tenemos que dar un cambio C. Hay una serie de tipos de billetes B0..B(N-1). El billete de tipo **Bk** tiene un valor **vk**. Podemos coger hasta **dk** billetes de ese tipo. Nuestro objetivo es alcanzar el cambio exactamente (si lo hay) con los menos billetes posibles.

Un ejemplo de problema práctico:

Queremos devolver un cambio de 7 euros.
Disponemos de los siguientes billetes:
Tipo 1: 4x 2 eur Tipo 2: 1 x 10 eur Tipo 3: 2 x 5 eur Tipo 4: 5 x 1 eur.

Se pide:
a) Diseñar un algoritmo que resuelva este problema.
b) Aplicarlo en papel al ejemplo de problema práctico propuesto y dar la solución óptima si la hubiera.

Solución:

a) Este ejercicio no se puede resolver con un algoritmo greedy porque no podemos encontrar una función de selección que nos garantice encontrar una solución, y que además si se encuentra sea la óptima. Esto es así porque en cada paso tomaremos ciegamente un billete determinado, que al final podría o no formar parte de la solución óptima (porque en un principio no lo podemos determinar), y ya no vamos a poder sacarlo porque greedy no se retracta de las decisiones tomadas.

Este ejercicio sí se puede, sin embargo, resolver con programación dinámica porque cumple el principio de optimalidad de Bellman, esto es, es descomponible en subestructuras optimales. Dicho con otras palabras, "dada una secuencia óptima de decisiones, toda subsecuencia de ella es, a su vez, óptima", y dicho de una forma menos formal, la solución óptima a un problema pequeñito de cambio Cpeque en el que disponemos de un número de billetes determinado va a ser formar parte de la solución óptima a un problema de más grande, Cgrande, en el que disponemos del mismo número de billetes más un billete de valor Cgrande-Cpeque.

En primer lugar vamos a decidir la representación del vector de decisiones a usar.

Usaremos un vector **x** que contendrá, en cada posición **xk**, el número de veces que cogemos el billete **Bk**. Por ejemplo, si disponemos de 4 tipos de billetes, el vector de decisión para un subproblema determinado podría ser {1,2,1,0}, lo que significaría que tomamos 1 billete de tipo 0, 2 billetes de tipo 1 y 1 billete de tipo 2.

En segundo lugar, definimos la función recurrente, que devolverá **un vector de decisión**:

```
                            { numbill(l,k-1,C)              si no cogemos el billete.
numbill(l,k,C) =  min sum   { numbill(l,k,C-1)              si no cogemos el billete (mismo tipo).
                            { (INC xk)numbill(l,k-1,C-vk)   si cogemos el billete (xk++)
                            {                                        (DEBEMOS PODER COGERLO)
                            { (INC xk)numbill(l,k,C-vk)     si cogemos el billete (xk++) (mismo tipo).
                            {                                        (DEBEMOS PODER COGERLO)
                            { 0                             si C = 0
                            { 0                             si k < l
                            { +INF                          si C < 0 (como minimizamos, garantiza que no se coge).
```

Sujeta a **Σxi*vi < C** y a que para cada i entre 0 y n-1, **xi <= di** (no cogemos más billetes de los que hay).

Obsérvese que la solución devuelta por esta función podría no ser completa (podría no alcanzarse el valor de cambio objetivo con un conjunto de billetes escogidos).

(en el examen no sería necesario detallar el contenido de esta página)

Paso a explicar el por qué de las cuatro opciones:

- Opción 1: numbill(l,k,C) = numbill(l,k-1,C)   si no cogemos el billete.

Esta opción viene a considerar el caso en el que la solución óptima para el problema actual es la misma que si no consideramos ningún billete del tipo actual. Esto es, nunca hemos cogido un billete del tipo actual y además en este paso tampoco.

- Opción 2: numbill(l,k,C) = numbill(l,k,C-1)   si no cogemos el billete (mismo tipo).

Esta opción viene a considerar el caso en el que la solución óptima para el problema actual es la misma que si, habiendo podido coger en pasos anteriores billetes del tipo actual, en este paso no lo hemos hecho. Obsérvese que hace falta reducir el cambio máximo sin reducir el número de elementos considerados, para considerar justamente el problema que incluye este tipo de billetes pero con un cambio un poquito más pequeño (porque ahora mismo estamos considerando el problema con cambio C).

- Opción 3: numbill(l,k,C) = (INC xk)numbill(l,k-1,C-vk)   si cogemos el billete (xk++)
  (DEBEMOS PODER COGERLO)

Esta opción viene a considerar el caso en el que la solución óptima para el problema actual es la misma que si antes no consideramos ningún billete del tipo actual pero en este paso cogemos uno. La condición de "DEBEMOS PODER COGERLO" indica que debemos tener disponibles al menos un billete más de este tipo de los que hay cogidos actualmente. La salida que proporciona la ejecución de esta función es la misma que numbill(l,k-1,C-vk) pero incrementando en uno en el vector de decisión la posición del billete que hemos cogido. Por ejemplo, si estamos en k=2 (considerando el billete tipo 2) y antes teníamos {0,1,0,0}, ahora tendremos {0,1,1,0}, es decir, se ha incrementado el número de veces que cogemos el billete 2.

- Opción 4: numbill(l,k,C) = (INC xk)numbill(l,k,C-1) si cogemos el billete (xk++) (mismo tipo).
  (DEBEMOS PODER COGERLO)

Esta opción viene a considerar el caso en el que la solución óptima para el problema actual es la misma que si, habiendo podido coger en pasos anteriores billetes del tipo actual, en este paso también lo hemos hecho. Obsérvese que hace falta reducir el cambio máximo sin reducir el número de elementos considerados, para considerar justamente el problema que incluye este tipo de billetes pero con un cambio un poquito más pequeño (porque ahora mismo estamos considerando el problema con cambio C). La salida que proporciona la ejecución de esta función es la misma que numbill(l,k,C-1) pero incrementando en uno en el vector de decisión la posición del billete que hemos cogido. Por ejemplo, si estamos en k=2 (considerando el billete tipo 2) y antes teníamos {0,1,1,0} (observar que ya estaba cogido al menos una vez), ahora tendremos {0,1,2,0}, es decir, se ha incrementado el número de veces que cogemos el billete 2.

Como nuestro objetivo es alcanzar el cambio exacto, vamos a hacer lo siguiente:

- Si mediante exactamente una de estas cuatro opciones alcanzamos el cambio exacto, vamos a coger esa opción, que proporciona una solución válida y completa.
- Si mediante más de una de estas cuatro opciones alcanzamos el cambio exacto, nos quedaremos con aquella que minimiza el número de billetes cogidos, o sea, la sumatoria de elementos del vector de decisión, porque lo que estamos buscando es justamente el mínimo.
- Si mediante ninguna de estas cuatro opciones alcanzamos el cambio exacto, como es la restricción más fuerte para obtener una solución válida para el problema, vamos a centrarnos en alcanzar nuestro objetivo. Tomaremos por tanto, de las cuatro opciones, aquella que más nos acerca, sin pasarnos, al cambio exacto que debemos proporcionar. Si hay empate, cogeremos la que use los menos billetes posibles.

Al final de la tabla, es decir, al resolver numbill(1,N-1,C), obtendremos un vector de decisión con la solución obtenida, que será una de dos:

- Si la solución no es completa, proporciona la mejor aproximación por debajo al cambio (por lo tanto, no hay solución al problema).
- Si la solución es completa (es decir, tenemos el cambio exacto), proporciona el vector de decisión con los números de billetes de cada tipo que debemos usar (por lo tanto, esa es solución al problema y es la óptima).

Por este motivo no hará falta hacer el recorrido hacia atrás para determina qué decisiones se tomaron. Se devuelve el resultado de la casilla final y ya está.

El algoritmo resultante sería el siguiente:

Como entrada vamos a recibir:

- N: Número de tipos de billetes.
- C: Cambio exacto a devolver.
- v: Vector de valores de los billetes.
- d: Vector de número de billetes disponibles.

Como salida vamos a proporcionar:

- x: Vector de decisión (número de billetes tomados).

Pseudocódigo:

Creamos una matriz de tamaño (N+1)*(C+1), en cada celda de la matriz va a haber un vector de decisión de tamaño N.

Inicializamos los casos base para que la primera fila de la matriz (no considerar ningún billete) y la primera columna de la matriz (cambio 0) sean todo 0.

Para cada celda de la matriz:
Generamos de las cuatro opciones, las que sean posibless (ojo, las que no se salgan de la tabla).
Comprobamos cuáles de ellas son válidas (cambio igual al cambio del subproblema que resolvemos).
Si hay alguna válida: De ellas, cogemos la que minimiza el número de billetes (sumatoria de elementos del vector).
Si no hay válidas: Cogemos, de todas, la que acercándose más por debajo al cambio a dar, minimiza el número de billetes (sumatoria de elementos del vector).
Ponemos en esa casilla el vector correspondiente a la opción que hemos tomado (incrementar si es necesario).

Nos fijamos en la última casilla:
Si es una solución válida, hay solución al problema y esa es la óptima.
Si no es una solución válida, no hay solución al problema.

b) Procedemos a resolver el problema en papel:
Queremos devolver un cambio de 7 euros.
Disponemos de los siguientes billetes:
Tipo 1: 4x 2 eur Tipo 2: 1 x 10 eur Tipo 3: 2 x 5 eur Tipo 4: 5 x 1 eur.

La tabla se inicializaría así:

B \ C	0	1	2	3	4	5	6	7
Hasta B0	{0,0,0,0}	{0,0,0,0}	{0,0,0,0}	{0,0,0,0}	{0,0,0,0}	{0,0,0,0}	{0,0,0,0}	{0,0,0,0}
Hasta B1	{0,0,0,0}							
Hasta B2	{0,0,0,0}							
Hasta B3	{0,0,0,0}							
Hasta B4	{0,0,0,0}							

Y se rellenaría así (ver desarrollo de algunos casos en las siguientes páginas), subrayo las válidas.

B \ C	0	1	2	3	4	5	6	7
Hasta B0	{0,0,0,0}	{0,0,0,0}	{0,0,0,0}	{0,0,0,0}	{0,0,0,0}	{0,0,0,0}	{0,0,0,0}	{0,0,0,0}
Hasta B1	{0,0,0,0}	{0,0,0,0}	{1,0,0,0}	{1,0,0,0}	{2,0,0,0}	{2,0,0,0}	{3,0,0,0}	{3,0,0,0}
Hasta B2	{0,0,0,0}	{0,0,0,0}	{1,0,0,0}	{1,0,0,0}	{2,0,0,0}	{2,0,0,0}	{3,0,0,0}	{3,0,0,0}
Hasta B3	{0,0,0,0}	{0,0,0,0}	{1,0,0,0}	{1,0,0,0}	{2,0,0,0}	{0,0,1,0}	{0,0,1,0}	{1,0,1,0}
Hasta B4	{0,0,0,0}	{0,0,0,1}	{1,0,0,0}	{1,0,0,1}	{2,0,0,0}	{0,0,1,0}	{0,0,1,1}	{1,0,1,0}

La solución final existe (es válida, llega a cambio 7) y es {1,0,1,0} (2 billetes).

(en el examen no sería necesario detallar el contenido de esta página ni de las siguientes)

Tipo 1: 4x 2 eur Tipo 2: 1 x 10 eur Tipo 3: 2 x 5 eur Tipo 4: 5 x 1 eur.

Para numbill(1,1,1):

- Opción 1: numbill(1,0,1) si no cogemos el billete.
  No válida/completa. Vale 0.
- Opción 2: numbill(1,1,0) si no cogemos el billete (mismo tipo).
  No válida/completa. Vale 0.
- Opción 3: (INC x0)numbill(1,0,-1) si cogemos el billete (DEBEMOS PODER COGERLO)
  No válida/completa. Aunque es posible coger el billete porque teníamos cogidos 0 de este tipo y tenemos hasta 4, el billete vale más que el cambio que tenemos que dar, así que no la podemos escoger.
- Opción 4: (INC x0)numbill(1,1,0) si cogemos el billete (mismo tipo). (DEBEMOS PODER COGERLO)
  No válida/completa. Aunque es posible coger el billete porque teníamos cogidos 0 de este tipo y tenemos hasta 4, el billete vale más que el cambio que tenemos que dar, así que no la podemos escoger.

Como ninguna de las posibles (la 1 y la 2) son válidas, cogemos la que nos acerca más al cambio a devolver, y en caso de empate, cogemos la que usa el menor número de billetes. En este caso nos empatan también, así que cogemos la 1 o la 2. Se puede ver que el vector de decisión para nuestro problema se iguala al que teníamos para la opción escogida.

Para numbill(1,1,2):

- Opción 1: numbill(1,0,2) si no cogemos el billete.
  No válida/completa. Vale 0.
- Opción 2: numbill(l,1,1) si no cogemos el billete (mismo tipo).
  No válida/completa. Vale 0.
- Opción 3: (INC x0)numbill(1,0,0) si cogemos el billete (DEBEMOS PODER COGERLO)
  VALIDA! El cambio alcanzado coincide con el objetivo. Podemos cogerlo. Incrementamos el número de veces del billete de tipo 1, por lo que tomamos el vector de decisión que teníamos para esta opción y le añadimos un billete de este tipo, queda {1,0,0,0}. Obsérvese que usamos 1 billetes y alcanzamos un cambio de 2.
- Opción 4: (INC x0)numbill(1,1,0) si cogemos el billete (mismo tipo). (DEBEMOS PODER COGERLO)
  VALIDA! El cambio alcanzado coincide con el objetivo . Podemos cogerlo. Incrementamos el número de veces del billete de tipo 1, por lo que tomamos el vector de decisión que teníamos para esta opción y le añadimos un billete de este tipo, queda {1,0,0,0}. Obsérvese que usamos 1 billetes y alcanzamos un cambio de 2.

Como dos de las opciones (3 y 4) son válidas, cogemos la que usa el mínimo de billetes (hay empate, así que cogemos una cualquiera). Se puede ver que el vector de decisión para nuestro problema se iguala al que teníamos para la opción escogida, considerando el incremento oportuno.

Para numbill(1,1,3):

- Opción 1: numbill(1,0,3) si no cogemos el billete.
  No válida/completa. Vale 0.
- Opción 2: numbill(l,1,2) si no cogemos el billete (mismo tipo).
  No válida/completa. Vale 1. Alcanzamos un cambio de 2 con el mismo vector de decisión {1,0,0,0}. **Importante:** se ve ahora por qué hace falta esta opción? Si no la tuviéarmos, no iríamos arrastrando el billete que cogimos en el paso anterior.
- Opción 3: (INC xk)numbill(1,0,1) si cogemos el billete (DEBEMOS PODER COGERLO)
  No válida/completa. Podemos cogerlo. Incrementamos el número de veces del billete de tipo 1, por lo que tomamos el vector de decisión que teníamos para esta opción y le añadimos un billete de este tipo, queda {1,0,0,0}. Obsérvese que usamos 1 billetes y alcanzamos un cambio de 2.
- Opción 4: (INC xk)numbill(1,1,1) si cogemos el billete (mismo tipo). (DEBEMOS PODER COGERLO)
  No válida/completa. Podemos cogerlo. Incrementamos el número de veces del billete de tipo 1, por lo que tomamos el vector de decisión que teníamos para esta opción y le añadimos un billete de este tipo, queda {1,0,0,0}. Obsérvese que usamos 1 billetes y alcanzamos un cambio de 2.

Como ninguna de las posibles (la 1, la 2, la 3 y la 4) son válidas, cogemos la que nos acerca más al cambio a devolver, y en caso de empate, cogemos la que usa el menor número de billetes. En este caso nos empatan también, así que cogemos la 2, la 3 o la 4. Se puede ver que el vector de decisión para nuestro problema se iguala al que teníamos para la opción escogida, considerando el incremento si corresponde.

Para numbill(1,1,4):

- Opción 1: numbill(1,0,4)   si no cogemos el billete.
  No válida/completa. Vale 0.
- Opción 2: numbill(l,1,3)   si no cogemos el billete (mismo tipo).
  No válida/completa. Vale 1. Alcanzamos un cambio de 2 con el mismo vector de decisión {1,0,0,0}.
- Opción 3: (INC x0)numbill(1,0,2) si cogemos el billete (DEBEMOS PODER COGERLO)
  No válida/completa. Podemos cogerlo. Incrementamos el número de veces del billete de tipo 1, por lo que tomamos el vector de decisión que teníamos para esta opción y le añadimos un billete de este tipo, queda {1,0,0,0}. Obsérvese que usamos 1 billetes y alcanzamos un cambio de 2.
- Opción 4: (INC x0)numbill(1,1,2) si cogemos el billete (mismo tipo). (DEBEMOS PODER COGERLO)
  VALIDA! El cambio alcanzado coincide con el objetivo. Podemos cogerlo. Incrementamos el número de veces del billete de tipo 1, por lo que tomamos el vector de decisión que teníamos para esta opción (que era {1,0,0,0}) y le añadimos un billete de este tipo, queda {2,0,0,0}. Obsérvese que usamos 2 billetes y alcanzamos un cambio de 4. **Importante:** se ve ahora por qué esta opción es importante? Nos permite escoger varios billetes del mismo tipo! En principio puede parecer igual si, en vez de definirla así "numbill(l,k,C) = (INC xk)numbill(l,k,C-1)" la hubiéramos definido así "numbill(l,k,C) = (INC xk)numbill(l,k,C-vk)", pero no sabemos qué podría estar pasando para un problema un poquito más pequeñito (a lo mejor, de las otras posiciones de la tabla nos vienen mejores soluciones para el tamaño C-1, y las tenemos que considerar también).

Como solo una de las opciones (la 4) es válida, cogemos esa.

Siguiendo este razonamiento voy a rellenar toda la primera fila. Si en algún momento se nos acabaran los billetes de este tipo, no podríamos seguir aplicando la opción 3 ni la opción 4.

Para la segunda fila, con el valor del billete 2 nunca vamos a añadirlo, así que se "heredaría" la solución de la primera fila siguiendo la función.

Para la tercera fila iba a ser igual que la anterior hasta 4, y en 5 cambiaría así:

Para numbill(1,3,5):

- Opción 1: numbill(1,2,5)   si no cogemos el billete.
  No válida/completa. Vale 2. Alcanzamos un cambio de 4.
- Opción 2: numbill(l,3,4)   si no cogemos el billete (mismo tipo).
  No válida/completa. Vale 2. Alcanzamos un cambio de 4 con el mismo vector de decisión {2,0,0,0}.
- Opción 3: (INC x2)numbill(1,2,0) si cogemos el billete (DEBEMOS PODER COGERLO)
  VALIDA! El cambio alcanzado coincide con el objetivo. Podemos cogerlo. Incrementamos el número de veces del billete de tipo 3 (x2), por lo que tomamos el vector de decisión que teníamos para esta opción ({0,0,0,0} porque C es 0) y le añadimos un billete de este tipo, queda {0,0,1,0}. Obsérvese que usamos 1 billetes y alcanzamos un cambio de 5.
- Opción 4: (INC x2)numbill(1,3,0) si cogemos el billete (mismo tipo). (DEBEMOS PODER COGERLO)
  VALIDA! El cambio alcanzado coincide con el objetivo. Podemos cogerlo. Incrementamos el número de veces del billete de tipo 3 (x2), por lo que tomamos el vector de decisión que teníamos para esta opción (que era {0,0,0,0}) y le añadimos un billete de este tipo, queda {0,0,1,0}. Obsérvese que usamos 1 billetes y alcanzamos un cambio de 5.

Como dos de las opciones (3 y 4) son válidas, cogemos la que usa el mínimo de billetes (hay empate, así que cogemos una cualquiera). Se puede ver que el vector de decisión para nuestro problema se iguala al que teníamos para la opción escogida, considerando el incremento oportuno.

El caso de numbill(1,3,6) es sencillito, se va a obtener una solución que no es válida.

Para numbill(1,3,7):

- Opción 1: numbill(1,2,7)   si no cogemos el billete.
  No válida/completa. Vale 3. Alcanzamos un cambio de 6.
- Opción 2: numbill(l,3,6)   si no cogemos el billete (mismo tipo).
  No válida/completa. Vale 3. Alcanzamos un cambio de 6 con el mismo vector de decisión {2,0,0,0}.
- Opción 3: (INC x2)numbill(1,2,2) si cogemos el billete (DEBEMOS PODER COGERLO)
  VALIDA! El cambio alcanzado coincide con el objetivo. Podemos cogerlo. Incrementamos el número de veces del billete de tipo 3 (x2), por lo que tomamos el vector de decisión que teníamos para esta opción ({1,0,0,0} porque C es 2) y le añadimos un billete de este tipo, queda {1,0,1,0}. Obsérvese que usamos 2 billetes y alcanzamos un cambio de 7.
- Opción 4: (INC x2)numbill(1,3,0) si cogemos el billete (mismo tipo). (DEBEMOS PODER COGERLO)
  VALIDA! El cambio alcanzado coincide con el objetivo. Podemos cogerlo. Incrementamos el número de veces del billete de tipo 3 (x2), por lo que tomamos el vector de decisión que teníamos para esta opción (que era {1,0,0,0}) y le añadimos un billete de este tipo, queda {1,0,1,0}. Obsérvese que usamos 2 billetes y alcanzamos un cambio de 7.

El razonamiento para los primeros casos de la cuarta fila lo dejo para que lo hagáis vosotros, en resumen, cada vez que falte uno para llegar al objetivo va a coger un billete de este tipo.

Para numbill(1,4,7):

- Opción 1: numbill(1,3,7)   si no cogemos el billete.
  VALIDA! El cambio alcanzado coincide con el objetivo. Vale 2. Alcanzamos un cambio de 7 con un vector de decisión {1,0,1,0}.
- Opción 2: numbill(l,4,6)   si no cogemos el billete (mismo tipo).
  No válida/completa. Vale 2. Alcanzamos un cambio de 6 con el mismo vector de decisión {0,0,1,1}.
- Opción 3: (INC x3)numbill(1,3,6) si cogemos el billete (DEBEMOS PODER COGERLO)
  No válida/completa. Vale 2 y alcanzamos un cambio de 6 con el mismo vector de decisión {0,0,1,1}.
- Opción 4: (INC x3)numbill(1,4,6) si cogemos el billete (mismo tipo). (DEBEMOS PODER COGERLO)
  VALIDA! El cambio alcanzado coincide con el objetivo.Vale 3 y alcanzamos un cambio de 7, el vector de decisión sería {0,0,1,2}.

Nuestras opciones válidas son la 1 y la 4. De ellas, cogemos la que minimieza el número de billetes (la 1, que vale 2).

Hemos acabado de rellenar la tabla. Ahora nos vamos a la última casilla. Es solución válida, por lo que hay solución válida para nuestro problema, con la que se alcanza el cambio que debemos dar exacto. Por tanto, nuestra solución es la óptima.

La solución es {1,0,1,0}, es decir, coger un billete de 2 y un billete de 5. Efectivamente, el cambio dado es 7 (el objetivo) y hemos usado solamente 2 billetes.

Programación dinámica, problema de cambio con varios billetes de cada tipo (limitados).
Autor: Luis Quesada Torres.

(VERSIÓN RESUMIDA)

Enunciado:

Tenemos que dar un cambio C. Hay una serie de tipos de billetes B0..B(N-1). El billete de tipo **Bk** tiene un valor **vk**. Podemos coger hasta **dk** billetes de ese tipo. Nuestro objetivo es alcanzar el cambio exactamente (si lo hay) con los menos billetes posibles.

Un ejemplo de problema práctico:

Queremos devolver un cambio de 7 euros.
Disponemos de los siguientes billetes:
Tipo 1: 4x 2 eur Tipo 2: 1 x 10 eur Tipo 3: 2 x 5 eur Tipo 4: 5 x 1 eur.

Se pide:
a) Diseñar un algoritmo que resuelva este problema.
b) Aplicarlo en papel al ejemplo de problema práctico propuesto y dar la solución óptima si la hubiera.

Solución:

a) Este ejercicio no se puede resolver con un algoritmo greedy porque no podemos encontrar una función de selección que nos garantice encontrar una solución, y que además si se encuentra sea la óptima. Esto es así porque en cada paso tomaremos ciegamente un billete determinado, que al final podría o no formar parte de la solución óptima (porque en un principio no lo podemos determinar), y ya no vamos a poder sacarlo porque greedy no se retracta de las decisiones tomadas.

Este ejercicio sí se puede, sin embargo, resolver con programación dinámica porque cumple el principio de optimalidad de Bellman, esto es, es descomponible en subestructuras optimales. Dicho con otras palabras, “dada una secuencia óptima de decisiones, toda subsecuencia de ella es, a su vez, óptima”, y dicho de una forma menos formal, la solución óptima a un problema pequeñito de cambio Cpeque en el que disponemos de un número de billetes determinado va a ser formar parte de la solución óptima a un problema de más grande, Cgrande, en el que disponemos del mismo número de billetes más un billete de valor Cgrande-Cpeque.

En primer lugar vamos a decidir la representación del vector de decisiones a usar.

Usaremos un vector **x** que contendrá, en cada posición **xk**, el número de veces que cogemos el billete **Bk**. Por ejemplo, si disponemos de 4 tipos de billetes, el vector de decisión para un subproblema determinado podría ser {1,2,1,0}, lo que significaría que tomamos 1 billete de tipo 0, 2 billetes de tipo 1 y 1 billete de tipo 2.

En segundo lugar, definimos la función recurrente, que devolverá **un vector de decisión**:

```
                            { numbill(l,k-1,C)              si no cogemos el billete.
numbill(l,k,C) =  min sum   { numbill(l,k,C-1)              si no cogemos el billete (mismo tipo).
                            { (INC xk)numbill(l,k-1,C-vk)   si cogemos el billete (xk++)
                            {                                       (DEBEMOS PODER COGERLO)
                            { (INC xk)numbill(l,k,C-vk)     si cogemos el billete (xk++) (mismo tipo).
                            {                                       (DEBEMOS PODER COGERLO)
                            { 0                             si C = 0
                            { 0                             si k < l
                            { +INF                          si C < 0 (como minimizamos, garantiza que no se coge).
```

Sujeta a **Σxi*vi < C** y a que para cada i entre 0 y n-1, **xi <= di** (no cogemos más billetes de los que hay).

Obsérvese que la solución devuelta por esta función podría no ser completa (podría no alcanzarse el valor de cambio objetivo con un conjunto de billetes escogidos).

El algoritmo resultante sería el siguiente:

Como entrada vamos a recibir:

- N: Número de tipos de billetes.
- C: Cambio exacto a devolver.
- v: Vector de valores de los billetes.
- d: Vector de número de billetes disponibles.

Como salida vamos a proporcionar:

- x: Vector de decisión (número de billetes tomados).

Pseudocódigo:

Creamos una matriz de tamaño (N+1)*(C+1), en cada celda de la matriz va a haber un vector de decisión de tamaño N.

Inicializamos los casos base para que la primera fila de la matriz (no considerar ningún billete) y la primera columna de la matriz (cambio 0) sean todo 0.

Para cada celda de la matriz:
Generamos de las cuatro opciones, las que sean posibless (ojo, las que no se salgan de la tabla).
Comprobamos cuáles de ellas son válidas (cambio igual al cambio del subproblema que resolvemos).
Si hay alguna válida: De ellas, cogemos la que minimiza el número de billetes (sumatoria de elementos del vector).
Si no hay válidas: Cogemos, de todas, la que acercándose más por debajo al cambio a dar, minimiza el número de billetes (sumatoria de elementos del vector).
Ponemos en esa casilla el vector correspondiente a la opción que hemos tomado (incrementar si es necesario).

Nos fijamos en la última casilla:
Si es una solución válida, hay solución al problema y esa es la óptima.
Si no es una solución válida, no hay solución al problema.

b) Procedemos a resolver el problema en papel:
Queremos devolver un cambio de 7 euros.
Disponemos de los siguientes billetes:
Tipo 1: 4x 2 eur   Tipo 2: 1 x 10 eur   Tipo 3: 2 x 5 eur   Tipo 4: 5 x 1 eur.

La tabla quedaría así:

B \ C	0	1	2	3	4	5	6	7
Hasta B0	{0,0,0,0}	{0,0,0,0}	{0,0,0,0}	{0,0,0,0}	{0,0,0,0}	{0,0,0,0}	{0,0,0,0}	{0,0,0,0}
Hasta B1	{0,0,0,0}	{0,0,0,0}	{1,0,0,0}	{1,0,0,0}	{2,0,0,0}	{2,0,0,0}	{3,0,0,0}	{3,0,0,0}
Hasta B2	{0,0,0,0}	{0,0,0,0}	{1,0,0,0}	{1,0,0,0}	{2,0,0,0}	{2,0,0,0}	{3,0,0,0}	{3,0,0,0}
Hasta B3	{0,0,0,0}	{0,0,0,0}	{1,0,0,0}	{1,0,0,0}	{2,0,0,0}	{0,0,1,0}	{0,0,1,0}	{1,0,1,0}
Hasta B4	{0,0,0,0}	{0,0,0,1}	{1,0,0,0}	{1,0,0,1}	{2,0,0,0}	{0,0,1,0}	{0,0,1,1}	{1,0,1,0}

La solución final existe (es válida, llega a cambio 7) y es {1,0,1,0} (2 billetes).