Diseño de Algoritmos

Problemas extra para actividades telemáticas

Tema 3. Algoritmos basados en Programación Dinámica

1. Tarificación postal
2. La serie de Stern

1. La tarificación postal

- En un determinado país se emiten *n* sellos diferentes de valores naturales positivos $s_1, s_2, \ldots, s_n$. Se quiere enviar una carta y se sabe que la correspondiente tarifa postal es T. ¿De cuántas formas diferentes se puede franquear exactamente la carta, si el orden de los sellos no importa?

  Guía:
  - Para solucionar este problema se puede definir la función:
    - formas(n, T) = número de formas de franquear T con n tipos de sellos

1. La tarificación postal

- Lo primero sería encontrar la definición de la solución.
  - En nuestro caso, puede ser una lista de valores de sellos.
    - Por ejemplo, para tarificar 57:
    - 1, 1, 5, 10, 20, 20
- Ahora hay que dar una definición recursiva del algoritmo.
  - Usaremos la función que nos ofrece la guía.
    - formas(i, j): usar los sellos del *1* al *i* para franquear la cantidad *j*
- Imaginemos que hay que franquear 25 y el sello más grande disponible (*i*) es de 50: **no podemos usar el sello de 50, nos pasamos**
  - En este caso habría que usar los sellos del 1 al i-1
    - si $s_i > j$ entonces formas(i, j) = formas(i-1, j)

1. La tarificación postal

- Imaginemos ahora que queremos franquear 25 y el sello más grande disponible es de 10. Podemos usarlo, pero también no usarlo (recuerda que nos piden formas de tarificarlo, unas serán usándolo y otras sin usarlo)
  - si $s_i \leq j$ entonces formas(i, j) = formas(sin usarlo)+formas(usándolo)
  - si $s_i \leq j$ entonces formas(i, j) = formas(i-1, j)+formas(i, j-s_i)

- Considerando los sellos con valores en $[s_1, s_i]$, para franquear una cantidad j podemos aplicar esta recurrencia:

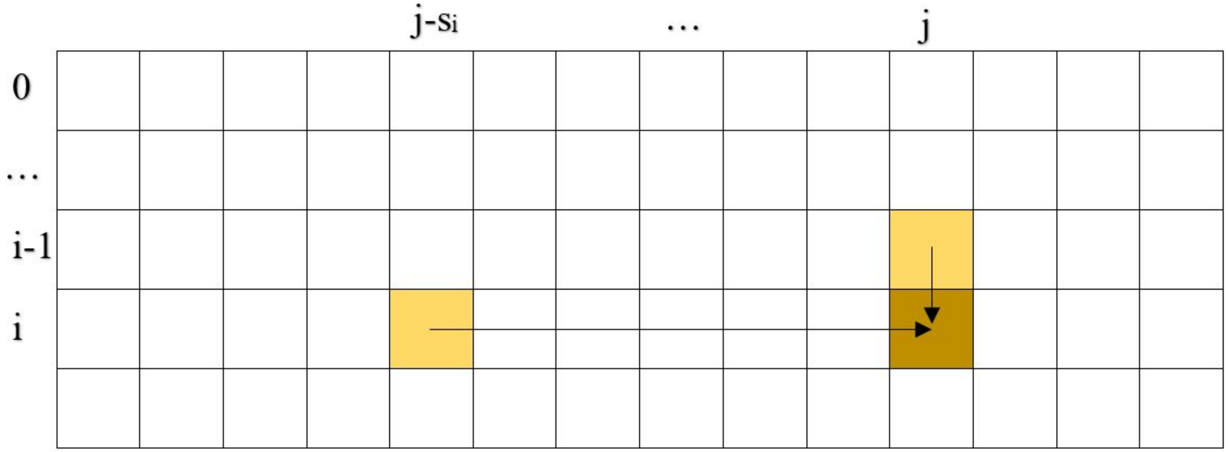
$$formas(i,j) = \begin{cases} formas(i-1,j) & si\ s_i > j \\ formas(i-1,j) + formas(i,j-s_i) & e.o.c. \end{cases}$$

1. La tarificación postal

- Ahora casos base y estructura de datos
  - Si no tengo sellos
    - $formas(0, j) = 0 \ \forall \ j \in [1, T]$
  - Si la cantidad a franquear es 0, solamente hay 1 forma: no poner ninguno
    - $formas(i, 0) = 1 \ \forall \ i \in [0, n]$
  - La estructura de datos puede ser una matriz,puesto que la función tiene dos dimensiones y, dado que hay índices 0, necesitamos una matriz de $(i+1)x(j+1)$
- Cómo la rellenamos?

1. La tarificación postal

- Cómo la rellenamos?
  - Tenemos ya la primera fila (fila 0 = 0) y la primera columna (columna 0 = 1)
  - Si no puedo poner el sello actual (si s_i>j) la solución es formas(i-1, j), es decir la casilla de arriba



  - Si puedo ponerlo, la solución es formas(i-1, j)+formas(i, j-s_i), es decir el resultado de sumar la casilla de arriba (no usarlo) y la casilla que está s_i columnas a la izquierda en la misma fila.

1. La tarificación postal

- Cómo la rellenamos?
  - Dos formas
    - De izquierda a derecha y de arriba abajo
    - De arriba a abajo y de izquierdas a derechas

Sellos = {1, 5, 10, 20} T = 12

	0	1	2	3	4	5	6	7	8	9	10	11	12
i=0	0	0	0	0	0	0	0	0	0	0	0	0	0
i=1	1	1	1	1	1	1	1	1	1	1	1	1	1
i=2	1	1	1	1	1	2	2	2	2	2	3	3	3
i=3	1	1	1	1	1	2	2	2	2	2	4	4	4
i=4	1	1	1	1	1	2	2	2	2	2	4	4	4

2. La serie de Stern

- Implementar un algoritmo basado en programación dinámica que calcule el término n-ésimo de la serie de Stern:

$$S(n) = \begin{cases} 0 & si\ n = 0 \\ 1 & si\ n = 1 \\ S(\frac{n}{2}) & si\ n\ es\ par \\ S\left(\frac{n-1}{2}\right) + S(\frac{n-1}{2} + 1) & si\ n\ es\ impar \end{cases}$$

Los primeros términos de la serie son: 0, 1, 1, 2, 1, 3, 2, 3, 1, 4, 3, 5, 2, 5, 3, 4, 1, 5, 4, 7, 3, 8, 5, 7, 2, 7, 5, 8, 3, 7, 4, 5, 1, 6, 5, 9, 4, 11, 7, 10, 3, 11, 8, 13, 5, 12, 7, 9, 2, 9, 7, 12, 5, 13, 8, 11, 3, 10, 7, 11, 4, 9, 5, 6, 1, 7, 6, 11, 5, 14, 9…