

PRÁCTICA 5

Ángela López López UO270318

EL PROBLEMA DEL CAMBIO:

Este problema aborda la forma de encontrar el número mínimo de monedas necesario para retornar un cambio previamente indicado. Es un tipo de [problema de la mochila](#).

Para resolver este problema se utiliza **programación dinámica**.

CAMBIO:

Clase que se encarga de realizar el problema del cambio. Tiene 2 constructores, uno que permite obtener el número de monedas distintas y el sistema monetario desde un fichero y otro que lo calcula aleatoriamente.

Esta clase cuenta con 2 métodos principales:

- **Int getNumMonedas(int cambio):** Calcula la cantidad de monedas mínimas necesarias para devolver el cambio especificado como parámetro.
- **int[] getMejoresMonedas():** Calcula qué cantidad de cada moneda se debe utilizar para poder obtener el cambio cumpliendo con el número mínimo de monedas calculado previamente.

Para poder utilizar estos métodos se trabaja con estos atributos.

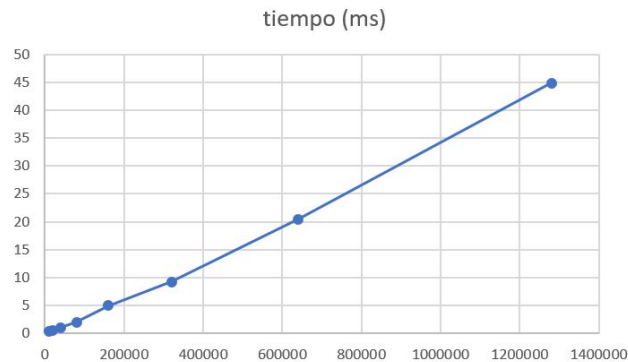
- n: Número de monedas distintas de las que dispone el sistema monetario.
- t: Vector que contiene las distintas monedas del sistema monetario.
- C: Matriz de enteros donde en cada componente [i,j] se calcula el número mínimo de monedas a devolver para una cantidad de monedas j, disponiendo de las monedas hasta t[i].
- marca: Matriz de booleanos que se utiliza para calcular cuántas monedas de cada tipo hay que dar en el cambio mínimo óptimo.
- cantidad: Número de cambio que buscamos obtener con la menor cantidad de monedas.

CAMBIO TIEMPOS:

n	tiempo (ms)
10000	0,332
20000	0,564
40000	1,024
80000	2,022
160000	4,914
320000	9,184
640000	20,468
1280000	44,842

Esta clase mide el tiempo que tarda en obtenerse el número de monedas mínimo a utilizar en un sistema aleatorio de una cantidad fija de monedas (100 en este caso) donde el cambio es n. En este caso solo es necesario obtener el número de monedas a devolver por lo que se llama al método **c.getNumMonedas(n);**

Como se puede apreciar en la tabla y la gráfica, tiene una complejidad lineal.



Para resolver el problema del cambio utilizamos el algoritmo Procedencia para la construcción de una matriz de booleanos “marca”, que nos permite conocer el número de monedas (y por lo tanto cuando se trata de n tipos de monedas diferentes y una cantidad L a conseguir, su complejidad es $O(n \cdot L)$).

Para obtener el número de monedas es necesario recorrer la matriz en orden descendente. Para ello hay que tener en cuenta lo que cuesta llegar a la primera fila y columna. Por lo tanto su complejidad **total** es $O(n + C[n, L])$.