

UNIDAD TEMÁTICA 1: Técnicas de Diseño de Algoritmos

TRABAJO DE APLICACIÓN 1 – Problema del “cambio de moneda”

Ejercicio #1 Solución ávida (15 minutos)

Escenario

El problema del cambio de moneda¹ es el conocido problema de descomponer cualquier cantidad dada, en monedas de un conjunto de valores dado, utilizando el **menor número total posible de monedas**.

Por ejemplo, tenemos que juntar \$23 y disponemos de monedas de \$1, \$2 y \$5.

La solución intuitiva a este problema consiste en usar la mayor cantidad posible de monedas del mayor valor, con el resto hacer lo mismo con la moneda del siguiente valor, y así sucesivamente hasta componer el importe deseado.

Para este ejemplo tomamos 4 monedas de \$5 para obtener los primeros \$20 pesos. Para los \$3 restantes usamos una moneda de \$2, y el último peso restante lo completamos con una moneda de \$1. Se usaron en total 6 monedas: 4 de \$5, 1 de \$2 y 1 de \$1.

La solución descrita es una solución típica de la técnica ávida (o voraz). Un posible algoritmo² que la implementa es:

int[] cambiarAvido (int importe, int[] monedas) //importe es la cantidad a descomponer y el arreglo “monedas” contiene los valores de monedas, con el menor valor en la posición cero.

Comienzo

```
int[] salida = nuevo arreglo de enteros, de longitud igual a “monedas”;  
desde i = monedas.largo-1 hasta 0 hacer  
    mientras monedas[i] <= importe  
        salida[i]++;  
        importe = importe - monedas[i];  
    fin mientras  
fin desde  
devolver salida;
```

Fin

De acuerdo al ejemplo planteado, si “importe” es 23 y el arreglo “monedas” es (1,2,5), el arreglo “salida” será (1,1,4) indicando que se usará una moneda de \$1, una moneda de \$2 y cuatro monedas de \$5

¹ Mark Allen Weiss, Estructuras de datos en Java, 4ta edición, página 322

² CAP4, página 144

Consigna

Estudia detenidamente el algoritmo planteado:

1. Identifica cada uno de los componentes de la técnica:
 - a. el conjunto de candidatos
 - b. Función solución
 - c. Función de selección
 - d. Función de factibilidad
 - e. Función objetivo
2. Prueba que el algoritmo escrito soluciona el problema planteado para cualquier conjunto de entrada, o diseña un contraejemplo que refute la corrección.
3. Estima el orden del tiempo de ejecución.

Entrega

Cada equipo elaborará un póster que mostrará al plenario y posteriormente se subirá una imagen del póster a la tarea correspondiente antes de la hora que se indique.