

Algoritmos voraces

Objetivos

- Implementar en C++ un algoritmo que sigue una estrategia voraz.
- Implementar algoritmos de ordenación.
- Trabajar en grupo.

Actividad (Grupos de 2 o 3 miembros)

En el problema de la mochila fraccionaria se disponen de un total de n objetos cada uno de ellos con un peso y un valor y una mochila (o recipiente) que es capaz de soportar un peso máximo M . El objetivo de este problema es encontrar la mejor selección de objetos a introducir en la mochila de forma que el valor que lleve sea máximo y no se sobrepase el peso máximo M , teniendo en cuenta que los objetos se pueden fraccionar.

El siguiente pseudocódigo, estudiado en las sesiones de teoría (Ejemplo 2 del Tema 4), resuelve el problema.

```
función Mochila(M:real+,O:&cadena[n], P:&real+[n], V:&real+[n]):[0,1][n]

    X:[0,1][n]  peso:real+  i:entero+

    X ← 0
    OrdenarDecreciente_VP(O,V,P)
    i ← 1  peso ← 0
    mientras peso < M y i ≤ n hacer
        si peso + Pi ≤ M
            Xi ← 1
            peso ← peso + Pi
        si no
            Xi ← (M-peso)/Pi
            peso ← M
        fsi
        i ← i + 1
    fmientras
    devolver X
ffunción
```

Implementa el algoritmo Mochila siguiendo las reglas de transformación de pseudocódigo vistas en la práctica 1. La implementación debe corresponderse con el pseudocódigo dado siguiendo las reglas de transformación. Recuerda que en este algoritmo los vectores empiezan en la posición 1 y acaban en la posición n , siendo n el número de objetos.

Implementa la función OrdenarDecreciente_VP para que ordene los vectores O, V, P según el valor por unidad de peso. El funcionamiento de esta función se trabajó en las sesiones de teoría. Puedes utilizar alguno de los algoritmos de ordenación vistos en la asignatura.

Crea un programa que pida por teclado el peso máximo de la mochila M , el número total de elementos n , los nombres de los objetos (vector O), sus pesos (vector P) y valores (vector V). El tipo de datos de los nombres de los objetos (vector O) es string. Se deberá hacer reserva de memoria dinámica para los vectores y no se utilizarán librerías adicionales a iostream.

Una vez introducidos todos los valores se resolverá el problema llamando a la función Mochila y se mostrará por pantalla la solución obtenida (objetos incluidos en la mochila y valor de la mochila). Ejecuta el programa para diferentes valores de M , n , pesos y valores. En la Figura 1 se muestra un ejemplo de ejecución del programa.

```
Introduce peso máximo de la mochila (M): 15
Introduce número de elementos (n): 3
Introduce nombres de los objetos:
    nombre objeto 1: azul
    nombre objeto 2: rojo
    nombre objeto 3: amarillo
Introduce pesos:
    peso objeto 1: 8
    peso objeto 2: 6
    peso objeto 3: 3
Introduce valores:
    valor objeto 1: 40
    valor objeto 2: 24
    valor objeto 3: 18

Objetos, pesos y valores
=====
(azul,8,40)
(rojo,6,24)
(amarillo,3,18)

Objetos introducidos:
=====
(amarillo,3,18)
(azul,8,40)
(rojo,6,24) 0.666667

Valor de la mochila: 74

Presione una tecla para continuar . . .
```

Figura 1: Ejemplo de ejecución del programa.