

Prácticas de Algorítmica.

3º de Grado en Ingeniería Informática. Curso 2023-2024.

Práctica 4.

Objetivos.

Con esta práctica se pretende que el alumno implemente dos algoritmos basado en la técnica de la Programación Dinámica los cuales proporcionan soluciones óptimas en cualquier caso.

El programa tendrá un menú con los apartados siguientes:

Apartado 1. Problema del cambio. (5 puntos).

Implementad el algoritmo de obtención del cambio con número mínimo de monedas o billetes para una cantidad entera de céntimos introducida por el usuario.

Se usarán los valores de las monedas y billetes del sistema monetario europeo. Los valores se tratan en céntimos para poder trabajar con números enteros, y se almacenarán en un fichero de texto denominado **“sistemamonetario.txt”** en orden decreciente que será leído por el programa. Este archivo se suministra con el resto del material.

Se suministra a clase Moneda para trabajar con los distintos tipos de moneda.

La matriz de estados que contiene el número óptimo de monedas para todos los estados posibles se guardará en un vector de vectores de enteros de la STL.

La solución se guardará en un vector de enteros de la STL en la que el elemento i-ésimo almacenará cuantas unidades se necesitan para la i-ésima moneda o billete del sistema monetario.

El sistema monetario se leerá del fichero citado anteriormente y se almacenará en un vector de la clase Moneda. El prototipo de esta función será:

`void cargarSistemaMonetario(vector <Moneda> &sistemaMonetario, const char *nombreFichero);`

La función que calcula el número de monedas óptimo para todos los estados posibles tendrá el siguiente prototipo:

`void cambio(int cantidad, vector <Moneda> &sistemaMonetario, vector<vector <int>>& matrizEstados);`

La función que calcula la solución a partir de la matriz de estados, tendrá el siguiente prototipo:

`void obtenerSolucion(vector<vector <int>>& matrizEstados, vector <Moneda> &sistemaMonetario vector<int> &solucion);`

La función que muestra la solución del problema tendrá el siguiente prototipo:

`void escribirSolucion(vector<int> &solucion, vector <Moneda> &sistemaMonetario);`

Notas importantes:

1. La matriz de estados será de $n \times N + 1$, siendo n el número de monedas/billetes del sistema monetario y N el cambio a conseguir.
2. Usar la indexación de C++ (empieza en 0 y termina en $n-1$) tanto en vectores como en matrices.
3. El elemento de la matriz de estados que proporciona el número de monedas/billetes de la solución final es el $matrizEstados(n-1, N)$.

Apartado 2. Problema de la mochila. (5 puntos)

Implementad el problema de la mochila. Los datos de entrada serán el volumen de la mochila y los materiales. El volumen de la mochila se introducirá por teclado y los materiales se almacenarán en un fichero de texto, con un registro para cada material, y se almacenará una etiqueta (1,2, 3, ..), su volumen y su precio por unidad de volumen. Se suministra el archivo **materialesmochila.txt** como ejemplo.

Se suministra a clase Material para trabajar con los distintos tipos de material.

Se suministra la clase MaterialUsado que se usará para guardar la solución. Dicha clase contendrá un objeto de la clase Material y el volumen usado de ese material.

La matriz de estados que contiene el costo máximo de la mochila para todos los estados posibles se guardará en un vector de vectores de doubles de la STL.

La solución se guardará en un vector de elementos de la clase MaterialUsado en la que el elemento i-ésimo almacenará el material i-ésimo que forma parte de la solución y el correspondiente volumen usado del mismo.

Los materiales disponibles se leerán del fichero citado anteriormente y se almacenará en un vector de la clase Material. El prototipo de esta función será:

void cargarMateriales(vector<Material> &materiales, const char *nombreFichero);

La función que calcula el coste óptimo para todos los estados posibles tendrá el siguiente prototipo:

void mochila(float volumenMochila, vector<Material> &materiales, vector<vector<double>>& matrizEstados);

La función que calcula la solución a partir de la matriz de estados, tendrá el siguiente prototipo::

void obtenerSolucion(vector<vector<double>>& matrizEstados , vector<Material> &materiales vector<MaterialUsado> &solucion);

La función que muestra la solución del problema tendrá el siguiente prototipo:

void escribirSolucion(vector<MaterialUsado> &solucion);

En la solución debe aparecer el volumen que se emplea de cada material y el coste final de la mochila. Probar el siguiente caso, con los siguientes valores de volumen con el fichero de texto suministrado, $V = 30, 90, 100, 135, 160, 200$.

Notas importantes:

- 1. La matriz de estados será de $n \times V + 1$, siendo n el número de materiales y V el volumen entero de la mochila.**
- 2. Usar la indexación de C++ (empieza en 0 y termina en $n-1$) tanto en vectores como en matrices.**
- 3. El elemento de la matriz de estados que proporciona el coste final de la mochila de la solución final es el $matrizEstados(n-1, V)$.**

Fecha de comienzo: 21 de noviembre.

Fecha máxima de entrega: 4 de diciembre(lunes).