

Identificación

- Nombre Autor: Vhanessa Cardona
- Identificación Autor: 201728751

Algoritmo de Solución

- **Explicación:** El algoritmo de solución presentado se basa en programación dinámica. Para este algoritmo se cuentan con 2 métodos *processProblem()* y *gain()*, cada uno se presenta más adelante con su respectivo contexto. En términos generales, el algoritmo calcula la ganancia si se elige cada uno de los proveedores y si se seleccionan los primeros i productos para producir; la selección de en qué cantidades producir se realiza con el método *gain()* (explicación más adelante)
- **Contexto del método *processProblem()*:**

| E/S | Nombre | Tipo | Descripción |
|-----|--------|--------------|-----------------------------------------------|
| E | n | nat | número de productos |
| E | k | nat | número de proveedores |
| E | a[n] | Array of nat | Kilos de madera para producir cada producto |
| E | p[n] | Array of nat | Precio de venta de los productos |
| E | m[k] | Array of nat | Kilos de madera de cada proveedor |
| E | c[k] | Array of nat | Costo de la madera de cada proveedor |
| S | pr | nat | Índice del proveedor que maximiza la ganancia |
| S | g | nat | Ganancia máxima |

Este método se desarrolla con programación dinámica, utilizando una matriz[n][k] para almacenar los valores de las ganancias de seleccionar el proveedor de la columna j y seleccionando solo productos entre la posición 0 e i .

$$gains[i][j] = \begin{cases} gain(i, j) & \text{si } i = 0 \\ \max(gains[i-1][j], gain[i, j]) & \text{si } i > 0 \end{cases}$$

Al finalizar el llenado de la matriz, la respuesta corresponde al máximo valor de la última fila de la matriz y el índice de la columna correspondiente.

Precondición:

$$|a| = n \wedge |p| = n \wedge |m| = k \wedge |c| = k \wedge n > 0 \wedge k > 0$$

- **Contexto del método *gain()*:**

| E/S | Nombre | Tipo | Descripción |
|-----|----------------|--------------|---------------------------------------------------------------------------------------------------------------------------|
| E | first_i | nat | Indice hasta el cual se van a considerar los productos |
| E | Index_supplier | nat | Indice del proveedor |
| E | n | nat | <i>número de productos</i> |
| E | k | nat | <i>número de proveedores</i> |
| E | a[n] | Array of nat | Kilos de madera para producir cada producto |
| E | p[n] | Array of nat | Precio de venta de los productos |
| E | m[k] | Array of nat | Kilos de madera de cada proveedor |
| E | c[k] | Array of nat | Costo de la madera de cada proveedor |
| S | g | nat | Ganancia que se tiene al seleccionar al proveedor con índice Index_supplier y considerando los primeros first_i productos |

Este método calcula el valor de la ganancia que se obtiene si se consideran los primeros i productos del array y si se selecciona el proveedor index_supplier. La forma de realizar este calculo consiste en añadir el número máximo de unidades del producto i, posteriormente el número máximo de unidades del producto i-1 y así hasta llegar al producto 0 o hasta que todos los kilos del proveedor ya se hallan utilizado. Cada vez que se añade una cantidad de unidades de un producto, se calcula el valor de venta total y se suma a la ganancia.

Precondición:

$$|a| = n \wedge |p| = n \wedge |m| = k \wedge |c| = k \wedge n > 0 \wedge k > 0$$

Postcondición:

$$\begin{aligned}
 g = & \left(+j | (first_i \geq j \geq 0) \wedge (diferencia \neq 0) : diferencia \right. \\
 & = diferencia - \left\lfloor \frac{kilosDisponibles}{a[j]} \right\rfloor * a[j] : \left\lfloor \frac{kilosDisponibles}{a[j]} \right\rfloor * p[j] \Big) \\
 & - c[index_{supplier}]
 \end{aligned}$$

Análisis de Complejidades Espacial y Temporal

- ***gain()***

Temporal: O(i) donde i son los primeros i productos

*Espacial: 6 * O(1) donde el 6 corresponde a 6 int*

- ***processProblem()***

*Temporal: O(n² * k) donde n es el # de productos y k el # de proveedores*

*Espacial: O(n * k) donde n es el # de productos y k el # de proveedores*

Comentarios Finales

La solución es en términos generales O(n³) lo cual se traduce en un buen desempeño