



INSTITUTO POLITÉCNICO NACIONAL
UNIDAD PROFESIONAL
INTERDISCIPLINARIA DE INGENIERÍAS
CAMPUS ZACATECAS (UPIIZ)



Mochila

Programa Académico: Análisis de Algoritmos

Profesor: Roberto Oswaldo Cruz Lieja

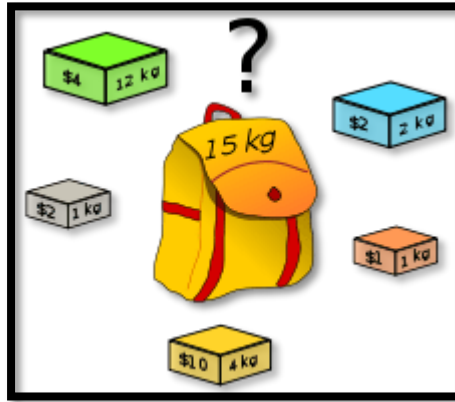
Alumno: Fernando Hipólito Vázquez Esparza

Asignación: Mochila

Fecha: 06/11/2019

Introducción

En este reporte veremos el problema de la mochila que básicamente, consiste en lo siguiente: tenemos una mochila con un tamaño máximo que nosotros le indicamos, también tenemos varios elementos que tienen un peso y un valor o beneficio.



El objetivo del programa es sacar la mejor combinación con mayor beneficio siempre y cuando no supere el peso de la mochila.

Desarrollo

Primero empezamos, con el hecho de usar el programa usado para el método burbuja, para la generación de costos y beneficios random para los artículos, es necesario tomar en cuenta que podemos definir cuantos artículos puede haber en la mochila, además de ver una pequeña matriz que muestra sus costos y beneficios por si solos.

Un link de esta clase:

<https://github.com/Alumnp/Analisis-de-algoritmos/blob/master/mochila/src/Mochila/ArticulosRandom.java>

Después podemos pasar a la clase que se llama artículos, la cual se encarga de recibir los datos del costo y beneficio que género la clase anterior, en pocas palabras, almacenamos los datos creados.

Link del código:

<https://github.com/Alumnp/Analisis-de-algoritmos/blob/master/mochila/src/Mochila/Articulos.java>

Después pasaríamos a la clase mochila, donde pasamos a llenar dichos datos e imprimirlos mediante ciclos, para mostrar la relación que hay entre ello.

Link del código: <https://github.com/Alumnp/Analisis-de-algoritmos/blob/master/mochila/src/Mochila/Mochila.java>

Por último, pasaríamos al método main que se encarga de llenar los datos de manera que podamos controlar cuando artículos hay en dicha mochila y además de poder asignar una rango del costo y beneficio de dichos datos

Link del main: <https://github.com/Alumnp/Analisis-de-algoritmos/blob/master/mochila/src/Mochila/Main.java>

Una pequeña captura de pantalla del resultado:

```
run:
Articulo 1  Costo: 2, Beneficio: 9
Articulo 2  Costo: 45, Beneficio: 42
Articulo 3  Costo: 59, Beneficio: 30
Articulo 4  Costo: 96, Beneficio: 52
Articulo 5  Costo: 49, Beneficio: 31
Articulo 6  Costo: 93, Beneficio: 88
90,72
2,9
45,42
59,30
96,52
49,31
93,88
BUILD SUCCESSFUL (total time: 0 seconds)
|
```

```
run:
Articulo 1  Costo: 41, Beneficio: 6
Articulo 2  Costo: 19, Beneficio: 17
Articulo 3  Costo: 55, Beneficio: 79
Articulo 4  Costo: 89, Beneficio: 13
Articulo 5  Costo: 19, Beneficio: 71
Articulo 6  Costo: 43, Beneficio: 46
94,85
41,6
19,17
55,79
89,13
19,71
43,46
BUILD SUCCESSFUL (total time: 0 seconds)
```

Podemos notar los cambios que hace la clase Artículos Random.

Conclusión.

En conclusión podemos ver que el uso de la mochila requiere de varios métodos que nos permita analizar el tamaño de los artículos, así como el acomodo que llevan, pese a que los datos son creados de forma aleatoria, son ordenamos respecto a un artículo, además de que no crea datos innecesarios que pueden comprometer a la mochila o la artículo.