Instituto Tecnológico de Costa Rica

PROYECTO DE INVESTIGACIÓN DE OPERACIONES

PROYECTO 1

KNAPSACK PROBLEM

Estudiantes

Jason Barrantes Arce
2015048456

Steven Bonilla Zúñiga
2015056296

Profesor

Francisco Torres Rojas

Modo Ejemplo:

Se resolverá un problema general por medio de diversos algoritmos que nos permitan encontrar una solución a ese problema. El problema que se nos plantea es sobre mochila.
Hay que llevar una mochila con capacidad de 15 (kilos, gramos) para un viaje. Tenemos una serie de objetos que podemos llevar, pero esos objetos tienen un respectivo peso y valor que será producido de manera aleatoria.

Restricciones:

- **Capacidad:** La mochila tendrá una capacidad de 15
- **Objetos:** Se generarán aleatoriamente 6 objetos.
- **Ci:** Varía entre $0 < Ci \leq 7$.
- **Vi:** Varía entre $0 < Vi \leq 20$.

Los tres algoritmos que vamos a implementar son:

- **Algoritmo de Programación Dinámica:** Algoritmo para el caso 0/1.
- **Algoritmo Greedy Básico:** Cada vez se escoge el objeto más valioso que quepa en lo que sobre de la mochila.
- **Algoritmo Greedy Proporcional:** En este se calcula el rendimiento definido como el valor del objeto dividido entre la capacidad que consume. Se escoge el de mayor rendimiento

En el caso de programación dinámica ya que nuestro objetivo es maximizar el valor que obtenemos, usamos la fórmula:

$$\text{MAX(Z)} = \sum_{i=1}^{n} x_i v_i$$

Que está sujeto a:

$$\sum x_i c_i \leq C$$

Con cada $x_i = 0$ o 1.

Se muestra a continuación la tabla de objetos con su respectivo costo (peso) y valor que fueron asignados aleatoriamente cumpliendo con las restricciones:

Nombre	**Costo**	**Valor**
Object 1	7	20
Object 2	6	14
Object 3	6	11
Object 4	4	11
Object 5	3	3
Object 6	4	12

1. Algoritmo Dinámico 0/1

Es un problema bidimensional, cuyo objetivo es maximizar la ganancia. En nuestro caso, queremos maximizar la cantidad de valores obtenidos por los objetos. Para solucionar el problema vamos a hacer uso de una tabla (m+1) x n, donde n es la cantidad de objetos disponibles y m es la cantidad de espacio disponible de la mochila. Como se menciona en las restricciones del problema $n = 6$ y $m = 15$, por lo que tendremos una tabla (16x6).

Fórmula Matemática

$$\text{MAX(Z)} = \sum_{i=1}^{6} x_i v_i$$

Sujeto a:

$$\sum x_i c_i \leq 15$$

En otras palabras tendremos:

$$\text{Z} = 20x_1 + 14x_2 + 11x_3 + 11x_4 + 3x_5 + 12x_6$$

Sujeto a:

$$7x_1 + 6x_2 + 6x_3 + 4x_4 + 3x_5 + 4x_6 \leq 15$$

Ahora proseguimos realizando la tabla dinámica.

X	1	2	3	4	5	6
0	0	0	0	0	0	0
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	3	3
4	0	0	0	11	11	12
5	0	0	0	11	11	12
6	0	14	14	14	14	14
7	20	20	20	20	20	20
8	20	20	20	20	20	23
9	20	20	20	20	20	23
10	20	20	20	25	25	26
11	20	20	20	31	31	32
12	20	20	25	31	31	32
13	20	34	34	34	34	34
14	20	34	34	34	34	37
15	20	34	34	34	34	43

El resultado es el siguiente: Las soluciones de X son las siguientes:

$$\text{Z} = (20 * 1) + (14 * 0) + (11 * 0) + (11 * 1) + (3 * 0) + (12 * 1)$$

$$\text{Z} = 43$$

$$X_1 = 1, X_2 = 0, X_3 = 0, X_4 = 1, X_5 = 0, X_6 = 1$$

Esta sujeto a:

$$(7 * 1) + (6 * 0) + (6 * 0) + (4 * 1) + (3 * 0) + (4 * 1) \leq 15$$

El algoritmo tarda aproximadamente: 0.035000 segundos en ejecutarse

2. Algoritmo Greedy Básico

Es un algoritmo que soluciona problemas que a primera vista parece ser óptimo. Es característico porque es muy sencillo de entender y explicar. Se escogen los objetos más valiosos que entren en lo que sobra de la mochila.

$$Obj_i = (Costo, Valor), i = 0...n$$

$$Obj_1 = (7, 20), Obj_2 = (6, 14), Obj_3 = (6, 11), Obj_4 = (4, 11), Obj_5 = (3, 3), Obj_6 = (4, 12)$$

Ahora proseguimos realizando la tabla greedy básico. Se muestra a continuación la tabla greedy basico con los resultados:

Objeto	**Peso**	**Valor**
Object 1	7	20
Object 2	6	14

$$\text{Z} = 34$$

El algoritmo tarda aproximadamente: 0.050000 segundos en ejecutarse

3. Algoritmo Greedy Proporcional

Es un algoritmo que soluciona problemas que a primera vista parece ser óptimo. Es característico porque es muy sencillo de entender y explicar. Se escogen los objetos más valiosos que entren en lo que sobra de la mochila, ya sea por su rendimiento.

$$Obj_i = (Costo, Valor), i = 0...n$$

$$Obj_1 = (7, 20), Obj_2 = (6, 14), Obj_3 = (6, 11), Obj_4 = (4, 11), Obj_5 = (3, 3), Obj_6 = (4, 12)$$

Ahora proseguimos realizando la tabla greedy proporcional. Se muestra a continuación la tabla greedy proporcional con los resultados:

Objeto	**Peso**	**Valor**	**Rend**
Object 6	4	12	3
Object 4	4	11	2
Object 2	6	14	2

$$Z = 37$$

El algoritmo tarda aproximadamente: 0.019000 segundos en ejecutarse