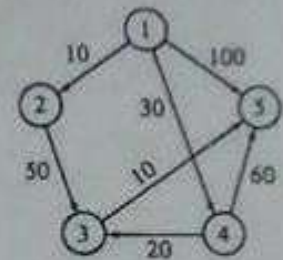


Ejercicio 1

Realice el seguimiento del algoritmo de Dijkstra sobre el grafo que se muestra a la derecha. Complete la tabla de seguimiento por cada paso del algoritmo, ya se ha completado el paso inicial. Suponga el vértice origen es el 1, S el conjunto de vértices ya incluidos en la solución, v el vértice seleccionado, D el array de distancias, y P el array de padres.



Iteración	S	v	D[2]	D[3]	D[4]	D[5]	P[2]	P[3]	P[4]	P[5]
0	{1}	-	10	∞	30	100	1	0	1	1
1										
2										
3										
4										

Ejercicio 2

Escriba en lenguaje JAVA un algoritmo utilizando la técnica Backtracking que resuelva el siguiente problema:

Dado un conjunto de n números enteros se desea encontrar, si existe, una partición en tres subconjuntos disjuntos, tal que la suma de sus elementos sea la misma.

Por ejemplo dado el conjunto $\{4,1,2,6,3,5,9\}$, existe una partición en tres conjuntos $\{4,6\}$, $\{5,3,2\}$, $\{9,1\}$ tal que cada uno de los tres conjuntos suma el mismo valor, 10.

Ejercicio 3

A Marcelo le gustan las facturas, pero también le preocupa su peso. Marcelo compró n facturas para comer, cada una con una cantidad de calorías c_j . Para mantener su peso, después de comer una factura Marcelo debe correr $c_j \cdot 2^j$ kilómetros, donde j es la cantidad que ya comió hasta el momento.

El problema para Marcelo es decidir en qué orden debe comer las facturas para al final en suma correr lo menos posible.

Por ejemplo, si Marcelo va a comer 3 facturas en total ($n=3$) con calorías 1, 2 y 3, y decidimos que coma primero una factura de 2 calorías, después una de 1 caloría y después una de 3 calorías; tendrá que correr 16 kilómetros ($2 \cdot 2^1 + 1 \cdot 2^2 + 3 \cdot 2^3 = 16$).

Dada la lista de facturas que comerá Marcelo, con las calorías de cada factura, ayude a Marcelo a decidir el orden que deberá comerlas mediante la implementación de un algoritmo con una estrategia Greedy, que deberá indicar la cantidad mínima de kilómetros que tiene que correr para mantener su peso.

Tener en cuenta que las facturas pueden ser comidas en *cualquier orden* y *todas* deben ser comidas.

Implementar todas las funciones necesarias para resolver el problema. La solución debe tener una complejidad computacional igual o menor a $O(n^2)$.

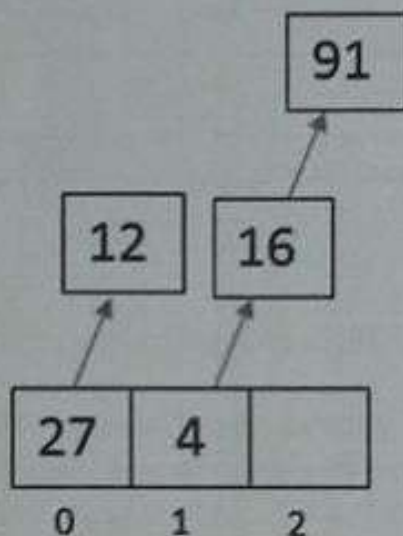
Ejercicio 4

Se tiene una estructura de Hashing separado con crecimiento lineal donde ya se ingresaron los siguientes elementos: 27, 4, 16, 12, 91.

El p de diseño es de 1.7; $M=3$, y la frontera f se encuentra en 0.

$$h(x) = x \bmod M$$

$$h'(x) = x \bmod 2 \cdot M$$



a) Ingresar el 33 y mostrar la estructura resultante. Cuál es el nuevo p de carga?

b) Ingresar el elemento 31 a la estructura resultante del punto a) y mostrar la estructura que se obtiene. Cuál es el nuevo p de carga?

Ejercicio 5

5) Contestar para cada afirmación, si es verdadera o falsa, justificando en cada caso. No se aceptarán respuestas sin justificación:

- A. El orden en que se insertan los elementos en un árbol binario de búsqueda no influye en la forma que adoptará el árbol.
- B. Si tuviera que escribir un algoritmo eficiente que busque un elemento en una estructura ordenada que guarde 500.000 elementos, elegiría guardarlos en una lista ordenada en vez de en una secuencia (array) ordenada.
- C. En una estructura de hashing cerrado es posible obtener un factor de carga mayor a 1.
- D. Un índice bitmap sirve para dominios de muchos valores diferentes, y sobre archivos que tienen gran cantidad de altas de nuevos elementos.