

Estructuras de Datos

PAO II 2021

Tarea Códigos de Huffman

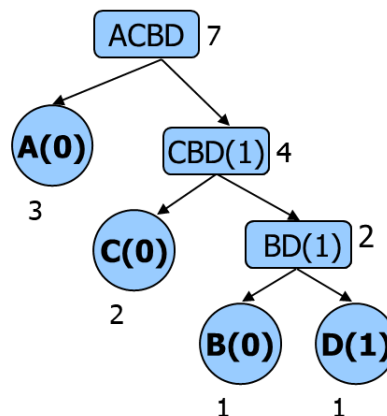
En esta tarea, usted implementará la codificación de Huffman utilizando el TDA BinaryTree.

1. Implemente el método **getFrequencies** que, dada una cadena de caracteres, retorne un diccionario representando la frecuencia con que cada carácter aparece en la cadena recibida.

Ejemplo: Dada la cadena **ABACCD**A, su método debe retornar el diccionario caracter-frecuencia mostrado a continuación:

A	3
B	1
C	2
D	1

2. Implemente el método **buildHuffmanTree** que, dado un diccionario representando la tabla de frecuencias de un conjunto de caracteres, retorne un árbol binario que permita codificar estos caracteres como códigos prefijos de Huffman. Por ejemplo, data la tabla de frecuencias anterior, su método debe retornar el siguiente árbol binario:



Recuerde que la construcción de este árbol se logra siguiendo los siguientes pasos:

PASO 1: Cada letra del diccionario, junto con su frecuencia y un bit, se introduce como contenido de la raíz de un árbol binario hoja, que es colocado en una cola de prioridad.

PASO 2: Removemos de nuestra cola de prioridad los dos árboles cuyo contenido tiene la menor frecuencia.

PASO 3: Los dos elementos removidos crean un nuevo elemento cuya frecuencia es la suma de las frecuencias de los dos anteriores y cuyo texto es la concatenación del texto de los dos removidos.

PASO 4: Este nuevo elemento se almacena en la raíz de un nuevo árbol binario hoja que tendrá como hijos izquierdo y derecho a los dos nodos removidos.

PASO 5: El nuevo árbol es añadido a la cola.

Se repite el proceso desde el **PASO 2** hasta que solo quede un árbol en la cola.

3. Implemente el método **getHuffmanCodes** que, dada una lista de caracteres (que no se repiten) y un árbol de codificación de Huffman, llena dos diccionarios que son recibidos vacíos como parámetros. Los diccionarios deben asociar cada caracter con su código correspondiente en dos maneras distintas. Por ejemplo, dados los caracteres **[A, B, C, D]** y el árbol de codificación del punto 2, su método debe llenar los dos diccionarios que recibe como parámetros de la siguiente manera:

Diccionario 1:
caracter-código

A	0
B	110
C	10
D	111

Diccionario 2:
código- caracter

0	A
110	B
10	C
111	D

Note que los dos diccionarios son muy similares. La única diferencia es que el primero tiene como claves a los caracteres mientras que las claves del segundo son los códigos.

4. Implemente el método **encode** que, dada una cadena de caracteres y un diccionario caracter-código, retorne la cadena recibida codificada. El valor retornado será un **String** de ceros y unos, que represente la cadena recibida. Por ejemplo, dada la cadena **ABACCD** y el Diccionario 1 del punto 3, el método **encode** debe retornar la cadena **011001010111**.
5. Implemente el método **decode** que, dada una cadena de unos y ceros que representa códigos prefijos de Huffman y un diccionario código-caracter, retorne la cadena decodificada. Por ejemplo, dada el **String 011001010111** y el Diccionario 2 del punto 3, el método **decode** debe retornar la cadena **ABACCD**.
6. Utilizando los métodos implementados arriba, escriba un programa principal que codifique el contenido de un archivo de texto. Para esto, su programa deberá:
- Leer todo el contenido del archivo como una sola cadena caracteres,
 - Generar la tabla de frecuencias con que aparecen los caracteres del texto leído (use el método **getFrequencies** para esto),
 - Construir el árbol de codificación de Huffman para la tabla de frecuencias generadas (con el método **buildHuffmanTree**).
 - Generar, utilizando el árbol de Huffman construido, los códigos correspondientes (mediante el método **getHuffmanCodes**).
 - Codificar el contenido del archivo iterando sobre la cadena correspondiente y creando una nueva de unos y ceros que resulte de concatenar los códigos prefijos generados (use para esto el método **encode** y el Diccionario1 que el punto anterior llenó).
 - Finalmente, guarde el texto codificado en un nuevo archivo de texto cuyo contenido será un conjunto de unos y ceros.
7. Utilizando los métodos que corresponda, escriba otro programa principal que decodifique el contenido de un archivo de texto que contiene unos y ceros. Para esto, usted deberá utilizar el método **decode** y el Diccionario 2 del punto e del ejercicio anterior. Guarde los resultados de su decodificación y verifique el contenido recuperado sea igual al archivo original.