Práctica Temas 9 y 10: La Codificación de Huffman

Estructura de Datos y de Algoritmos. Grupos GIC-A y GIS-B. Curso 2015-2016

Es posible reducir significativamente el número de bits requeridos para representar un texto si, en lugar de emplear un código de longitud fija como el ASCII, se emplea un código de longitud variable. En este caso, el número de bits requeridos puede variar de carácter en carácter. El objetivo es codificar los caracteres que aparecen más frecuentemente usando cadenas de bits más cortas. Sin embargo, cuando el código es de longitud variable se necesita algún método para determinar los bits de inicio y de fin de un código. Una forma de garantizar que una cadena de bits codificada se corresponde con una única secuencia de caracteres es asegurar que ningún código aparece como parte inicial de otro.

La técnica de los códigos de Huffman nos permite construir códigos sin prefijos, que además serán óptimos, en el sentido de que se elegirán menores longitudes de código para aquellos caracteres que aparezcan un mayor número de veces. El método de compresión/decompresión de datos consiste en lo siguiente:

- Calcular las frecuencias de todos los caracteres de un mensaje dado (tabla de frecuencias).
- Construir el "árbol de Huffman" a partir de la tabla de frecuencias.
- Construir una tabla de códigos de longitud variable utilizando el "árbol de Huffman".
- Una vez obtenidos los códigos de cada carácter se codifica el mensaje.
- El mensaje codificado se decodifica de forma directa usando el árbol de Huffman.

Un "árbol de Huffman" es un árbol binario que, o bien es un nodo hoja que contiene un carácter y su frecuencia, o bien está construido a partir de otros dos árboles de Huffman A1 y A2, hijo izquierdo y derecho respectivamente y como frecuencia de la nueva raíz la suma de las frecuencias de A1 y A2. El código de un carácter viene dado por el camino desde la raíz hasta la hoja que contenga al carácter, de forma que para cada bit del código, un "0" significa ir al hijo izquierdo, y un "1" significa ir al hijo derecho. El árbol se construye de forma que los códigos más cortos se corresponden con caracteres más frecuentes. Para esto, se utiliza una "cola con prioridad", que inicialmente contendrá los árboles hoja de la tabla de frecuencias. Mientras haya mas de un árbol en la cola, saldrán los dos árboles menores y se insertará uno nuevo construido a partir de estos dos. Tendremos definida una relación de orden respecto a los árboles (se ordenan de forma directa comparando las frecuencia de los nodos raíz).

Se pide diseñar una aplicación capaz de realizar la codificación de ficheros de texto utilizando la técnica descrita, y capaz de decodificar ficheros previamente codificados con dicha técnica, obteniéndose así un fichero con el texto original. Más concretamente:

1. Implementa la clase *HuffmanCodes*, la cual implementa métodos para construir la tabla de frecuencias a partir del texto de entrada, para construir el árbol de Huffman a partir de la tabla de frecuencias, para construir la tabla de códigos a partir del árbol de Huffman, para codificar el texto y para decodificarlo.

2. Implementa la clase *HuffmanTree* que representa a los árboles de Huffman. Las únicas operaciones necesarias son: (1) Un constructor de un árbol hoja, (2) un constructor de un árbol a partir de dos árboles dados, y, (3) una operación que determine si un árbol es o no una hoja. La representación puede usar nodos enlazados (al estilo de la clase *Arbin*) o bien usar herencia, en cuyo caso se implementaría una clase abstracta *HuffmanTree*, y dos clases concretas *HuffmanLeaf* y *HuffmanBranch*.

3. Implementar una función principal que presente un menú con dos opciones para: (1) Codificar un fichero (extensión .txt) produciendo el fichero codificado (extensión .huf), el cual debe incluir al principio la información para poder ser decodificado; (2) Decodificar un fichero .huf produciendo el .txt correspondiente.