

Algoritmo de Huffman
Análisis por Juárez Cruz Oscar Daniel
Análisis de Algoritmos 3CV6

Descripción: El algoritmo de Huffman tiene como propósito comprimir un archivo para reducir su espacio. Esta compresión la realiza conociendo la frecuencia de los símbolos que aparecen en un archivo. Con base en estas frecuencias se ordenan de menor a mayor y genera un árbol para obtener un diccionario donde asigna un identificador a cada símbolo como en el siguiente ejemplo:

Palabra: ALGORITMO

Símbolo	Frecuencia
A	1
L	1
G	1
R	1
I	1
T	1
M	1
O	2

Memoria estimada:

El archivo de n bytes + un nuevo archivo de $\%n$ bytes. En el peor de los casos utilizaríamos $2n$ bytes pero no más.

Orden computacional:

$$O(2n+m^2)$$

Donde m es igual al número de símbolos diferentes que aparecen en el documento. En nuestro ejemplo de "ALGORITMO" m sería 8. Está al cuadrado debido a que una vez que se hacen las frecuencias se ordenan, en este caso se contempla burbuja para el ordenamiento, en caso de utilizar algún otro la complejidad puede reducirse.

Algoritmo

```
CLASS Element{
    char element
    int aparicion
}

FUNCION comprobarDiccionario(List lista, Element element){
    IF( element.exists( lista ) )
        element.aparicion++
    ELSE
        lista.add( element )
}

MAIN{
    f = FILE( archivo.txt )
    Lista<Element> list
    MIENTRAS c = next( f ) HACER
        comprobarDiccionario( list, c )
    FIN
    ORDENAR( list )
    arbol = CREARARBOL( list )
    codigos = OBTENERCODIGOS( arbol )
    REEMPLAZARXCODIGOS( archivo.txt )
}
```

DESCRIPCIÓN de FUNCIONES OMITIDAS:

CREARARBOL := Crear un arbol con los valores de frecuencias en las hojas, inicialmente se propone que Huffman autocomplete si es necesario, posteriormente puede hacerse una mejor sin autocompletar las hojas.

OBTENERCODIGOS := Genera un nuevo diccionario con base en los recorridos del árbol otorgando así un nuevo símbolo (más pequeño en bits) a cada carácter que apareció en el algoritmo y el cual nos servirá para comprimir y descomprimir el archivo.

REEMPLAZARCODIGOS := Esta función deberá leer el archivo original mientras escribe el nuevo con los códigos obtenidos en el diccionario en la función OBTENERCODIGOS.