

Códigos de Huffman

Hacer esta práctica fue muy interesante ya que en mi vida había programado con C++, sin embargo, si tengo conocimientos en C así que solo se me dificulto un poco pasar el archivo Huffman.cpp a archivos Java que realizaran la misma tarea, solo tuve una complicación y fue con el método decodificar y es que hacia el mismo método que aparece en Huffman.cpp me salía del tamaño del String, para solucionar eso decidí comparar otras implementaciones de Huffman en Java que hay en internet y el resultado fue lo siguiente:

```
public String Decodificar(String s) {  
    System.out.println("LEN(s) " + s.length() + " " + s);  
    String r = "";  
    HuffmanNodo p = arrel;  
    for (int i = 0; i < s.length(); i++) {  
        int j = Integer.parseInt(String.valueOf(s.charAt(i)));  
        if (j == 0) {  
            p = p.fe;  
            if (p.fd == null && p.fe == null) {  
                r += p.c;  
                p = arrel;  
            }  
        }  
        if (j == 1) {  
            p = p.fd;  
            if (p.fd == null && p.fe == null) {  
                r += p.c;  
                p = arrel;  
            }  
        }  
    }  
    return r;  
}
```

```
}
```

En este código vamos recorriendo la cadena codificada carácter por carácter y vamos recorriendo el árbol de Huffman generado, así al final, obtendremos la cadena decodificada, es otra forma de poder decodificar el resultado.

Ahora de tener el código en donde se implementa el algoritmo de Huffman, se hizo la clase ManejoArchivo que es donde se lee, escribe y se comparan los archivos, los métodos para leer, escribir y comparar también se hicieron en el archivo Huffman.cpp pero en este se implementaron como métodos.

Algo importante a mencionar es que se necesitan tener los dos archivos generados, es decir, Codificacion Secuencia Genomica 1 C++ y Codificacion Secuencia Genomica 1 Java para que no salga ningún error en los dos casos, sin embargo, gracias a que se puede manejar ese error se muestra en pantalla el siguiente mensaje: “No se pudo abrir un archivo”, esto con la finalidad de que el usuario se dé cuenta de que hay un archivo faltante, aun a pesar de ese error se genera el archivo con la codificación y ahora si se ejecuta su contraparte esta funcionaria correctamente, es decir, si estamos en Huffman.java, el cual es el main, y no tenemos el archivo codificado de C++ se generara el archivo codificado usando Java, pero si después vamos a C++ y ejecutamos Huffman.cpp este si va a poder comparar ya que ya existen los dos archivos, esto también funciona al revés, es decir, de C++ a Java.

Un último punto para mencionar es que se creó en ambos casos un método llamado Frecuencia, en el cual se calcula la frecuencia de cada carácter que se encuentre en la cadena a codificar y se le asigna su clave y valor cada uno. El cálculo de la frecuencia es sencillo. Se toma el número de veces que aparece un mismo carácter en la cadena y se divide en la longitud de la cadena, así de esta forma obtenemos con qué frecuencia aparece cada carácter en la cadena, esto es necesario ya que no todas las cadenas son iguales y no todos los caracteres tendrán la misma frecuencia para otra cadena.

En conclusión, el algoritmo de Huffman nos permite mandar mensajes sin alto coste de tiempo, es aplicable en muchos campos en especial en donde se tienen que mandar mensajes con grandes datos, pero de manera rápida, un ejemplo, es con los telescopios satelitales estos se encuentran a grandes distancias y los datos que deben de mandar son considerablemente grandes, por ello el uso de Huffman es primordial ya que para mandar un mismo mensaje el tiempo que este se va a tardar sería diferente si no usáramos Huffman y nos damos cuenta de que este algoritmo voraz cumple con su objetivo de obtener una solución óptima al problema de mandar mensajes de manera rápida.

A continuación, se muestra la ejecución de Huffman en Java Imagen 1 y en C++ Imagen 2. También se anexa una captura de pantalla del archivo con la decodificación correspondiente en Imagen 3.

Imagen 1. Ejecución Huffman Java.

[illegible]

Imagen 2. Ejecución Huffman C++ (Se recortaron partes de la salida, ya que esta es muy grande).



Sanchez Mendez Edmundo Josué
Análisis de Algoritmos