

# Proyecto 3: Construcción de Árbol de Huffman

**Miércoles 16 de Octubre**

## I. DESCRIPCIÓN GENERAL

Este programa procesará el archivo de frecuencias generado en el proyecto anterior y creará un árbol de Huffman con dichos datos. Este árbol debe ser desplegado como un árbol binario en un documento  $\text{\LaTeX}$  y también se deben generar archivos “.h” con las estructuras de datos en C necesarias para que programas posteriores compriman y descompriman información con este árbol. Es muy importante el manejo de argumentos de la línea de comando para este proyecto. Toda la programación debe hacerse en Lenguaje C sobre Linux.

## II. PRELIMINARES

Tienen que investigar que `packages` hay disponibles para dibujar árboles binarios. Debe haber flexibilidad para poner información en los arcos y en los nodos hojas o intermedios.

También deben planificar que estructuras de datos necesitarían programas posteriores que compacten y descompacten información siguiendo los árboles de Huffman.

## III. ARGUMENTOS

Su programa (identificado como `<prog>`) se invocará de esta manera:

```
<prog> <freq-file> <argumentos>
```

Donde `<freq-file>` es un archivo de frecuencias generado con el proyecto previo. Su programa debe manejar apropiadamente los casos donde este archivo no exista.

Los `<argumentos>` pueden ser:

- `-h`: se crean archivos *header* con la terminación `.h`. Estos archivos contienen declaraciones e inicializaciones en C de las estructuras de datos necesarias para compactar y descompactar información siguiendo el árbol de Huffman generado con los datos de entrada. Estos archivos serán insertados con `#include` en el código C de proyectos posteriores.
- `-T`: se mostrará una tabla con la tira de bits correspondiente a cada caracter.
- `-t`: se genera un gráfico “básico” con el árbol de Huffman. Solo se mostrarán los caracteres en las hojas y la estructura del árbol binario, sin etiquetar los arcos.
- `-c`: muy similar a la opción `-t` pero incluyendo una asignación de ceros y unos en los arcos del árbol.
- `-f`: si se solicitó alguna de las opciones que despliegan el árbol, cada nodo hoja o intermedio mostrará la

probabilidad asociada usando 3 decimales. Si no se va a dibujar el árbol esta opción no hace nada.

- `-0`: se procesarán los 256 caracteres ASCII para generar el árbol Huffman. Es posible que muchos de estos tengan frecuencia cero. En caso de no estar esta opción, sólo se consideran aquellos cuya frecuencia sea estrictamente mayor a 0. La presencia de esta opción desactiva las opciones `-t`, `-c`, `-f`.

Los argumentos pueden aparecer en cualquier orden y cualquiera podría faltar. El programa debe manejar apropiadamente cualquier combinación inválida.

## IV. SALIDA

Si el usuario solicitó el árbol (`-t -c -f`) y/o la tabla de códigos (`-T`) su programa debe crear un documento  $\text{\LaTeX}$  que presente esta información. Aparte de crear el archivo “.tex” se invocará internamente<sup>1</sup> al programa `pdflatex` para que cree el PDF y al programa `evince` para que lo despliegue (en modo presentación). Su programa debe manejar apropiadamente los caracteres no desplegables (mostrar su código ASCII hexadecimal).

Cualquier despliegue de árbol, sólo se hará si hay 10 o menos caracteres.

**Trabajo extra opcional 1:** desplegar el árbol con 20 o más caracteres.

El contenido y formato de los archivos `.h` queda a criterio de cada grupo, pero los proyectos posteriores incluirán estos archivos en tiempo de compilación para generar compactadores y descompactadores.

## V. REQUISITOS INDISPENSABLES

La ausencia de uno solo de los siguientes requisitos vuelve al proyecto “no revisable” y recibe un 0 de calificación inmediata:

- La colaboración entre grupos se considera fraude académico.
- Todo el código debe estar escrito en C (no C++).
- El proyecto debe compilar y ejecutar en Linux.
- La presentación debe ser de mucha calidad.
- No debe dar “Segmentation Fault” bajo ninguna circunstancia.
- La única interacción válida con el programa es por medio de argumentos de línea de comando.

---

<sup>1</sup>`usar system()`

**Trabajo extra opcional 2:** hacer la demostración en una máquina que levante Linux de manera real (puede ser dual), es decir no usar máquinas virtuales.

## VI. FECHA DE ENTREGA

Revisiones a las 9:30 am el **Miércoles 16 de Octubre**.  
Mande además un `.tgz` con todo lo necesario (fuentes, make-file, readme, etc.) a `torresrojas.cursos@gmail.com`.  
Ponga como subject: A.A. - Proyecto 3 - Fulano - Mengano - Sutano, donde Fulano, Mengano y Sutano son los 3 miembros del grupo.