

## Actividad 8: desarrollo backend del algoritmo de Hoffman

30/04/2024

De Santiago Rodríguez Armando.

222362658

### Introducción:

El desarrollo de esta actividad se basa en conocer el como adaptar el código de un compañero para el desarrollo del backend con el fin de desarrollar un algoritmo de compresión y descompresión con los arboles de Huffman mediante el uso de códigos binarios.

### Desarrollo:

**Nodo:** Esta función fue creada para el manejo de nodos al momento de adaptarlos al árbol

**contruir\_arbol:** Esta función crea un árbol de Huffman asignando valores binarios a cada uno de los caracteres presentes en el archivo .txt.

**asignar\_codigos:** Esta función asigna binarios a cada carácter en el árbol de Huffman, la asignación de esto depende de la variable global llamada contador\_letras que va contando la cantidad de caracteres del mismo tipo y agregándola a un diccionario, el carácter con mayor cantidad de letras del mismo tipo presente en el txt se toma como primero asignándole el valor binario de 0.

**comprimir\_A:** Esta función comprime el archivo txt utilizando los binarios proporcionados y lo convierte en un archivo .bin.

**descomprimir\_A:** esta función descomprime el archivo .bin comprimido usando un árbol de Huffman y lo convierte nuevamente en un archivo .txt.




### Funciones de apoyo tomadas del Frontend:

**Leer:** Esta función lee los elementos del archivo fuente.

**Conteo:** Esta función ordena el diccionario especificado de mayor a menor y cuenta cada uno de los caracteres presentes en el archivo txt. NOTA: Agregue una pequeña función para ordenarlos de mayor a menos ya que no estaba presente en el Frontend.

## Resultados:

El código es capaz de comprimir y descomprimir archivos txt. La tasa de compresión depende de la distribución de caracteres en el archivo original. En las pruebas realizadas observe que al momento de descomprimir hay un ligero margen de peso en el archivo original (602KB) y en el que se descomprimió (591KB).

 gulliverTravels.txt	30/04/2024 06:12 p. m.	Documento de te...	602 KB
 gulliverTravelscomprimido.bin	30/04/2024 07:33 p. m.	Archivo BIN	327 KB
 gulliverTravelscomprimidodescomprimid...	30/04/2024 07:34 p. m.	Documento de te...	591 KB

## Conclusión:

El desarrollo de esta actividad me pareció muy complejo al inicio, ya que a pesar de que el código proporcionado estaba bien estructurado y comentarios, me resulto confuso el entender la lógica usada, y una vez que comprendí la lógica me puse a investigar el uso de los nodos en Python para el manejo de los árboles de Huffman, ya que desconocía si era similar el manejo de nodos en c++ con Python. También me resulto confuso al inicio el intentar convertir los caracteres ordenados en binario, al principio se me ocurrió utilizar la función "bin" de Python para convertir el diccionario ordenado de mayor a menor a binario, pero recordé que tenía que usar el uso de los arboles de huffman para realizar su conversión. Otro problema que tuve fue al momento de querer implementar un formato grafico para que se vean los nodos, pero no lo logre, ya que me aparecían las direcciones de los nodos, no los caracteres, así que omití esa parte.