UPC: CC182 - Algoritmos y Estructura de Datos
Trabajo Practico

Profesor: Todos
Sección: CC182
Semestre 1, 2021

23/06/2021

1. Instrucciones

- El lenguaje de programación será C++.

2. Enunciado: CRIPTO-UPC

En el departamento de UPC-Labs se ha realizado un modelo matemático para encriptar archivos.

Se solicita:

- Construir una estructura de datos (árbol binario) que pueda contener la codificación de los caracteres de un archivo.
- A partir de la estructura de datos generada, en el punto anterior, reconstruir un mensaje a partir de cualquier código que le sea entregado.

2.1. Encriptado

El proceso de encriptado genera un árbol binario para representar la codificación de los caracteres de un archivo (Vease el anexo 3.1 para mejor entendimiento).

Los pasos a seguir son:

1. Liste los caracteres del archivo con sus respectivas ocurrencias (i.e.: la cantidad de veces que cada caracter se repite en el archivo, desconsidere saltos de línea).
2. Calcule la probabilidad de ocurrencia de un caracter, y ordene los caracteres en función decreciente de su probabilidad. En caso de haber dos caracteres con la misma probabilidad de preferencia al caracter cuyo valor ASCII es menor.

$$probabilidad(caracter) = \frac{ocurrencia(caracter)}{TotalCaracteres} \tag{1}$$

3. Cada caracter tendrá asociado un código de 0's y/o 1's, que inicialmente estará vacío.

4. Divida la lista actual, de caracteres, en dos sublistas de la forma mas equilibrada posible, mantenga la primera lista como la de mayor porcentaje en caso de no tener un equilibrio pleno. Note que cada división de una lista genera dos hijos a un nodo.

5. Anexe a los códigos de los caracteres en la primera lista el valor 0, y a los de la segunda lista el valor 1.

6. Para cada una de las sublistas obtenidas repita los pasos 4 y 5, mientras estas contengan mas de un caracter.

2.2. Desencriptado

Dado un árbol binario representando la codificación de los caracteres de un alfabeto la decodificación es un proceso trivial. (Vease el anexo 3.2 para mejor entendimiento).

Seguimos los siguientes pasos:

1. A cada valor (0 o 1) en el archivo descendemos por la rama correspondiente (*i.e.*: rama izquierda si el valor leído es 0, rama derecha si el valor leído es 1).

2. Repita el paso 1 hasta llegar a un nodo hoja, en dicho caso imprima el caracter del nodo hoja.

3. Repita los pasos 1 y 2 hasta acabar de leer el código recibido.

2.3. Duración y Entrega

1. La solución debe ser enviada **únicamente** en un archivo .cpp o .txt.

2. La solución debe contener el código funcionando sin errores.

3. El nombre del archivo solución debe estar conformado por el primer nombre del alumno y el primer apellido del alumno (e.g.: FulanoMejia.txt o MenganoDelSolar.cpp).

2.4. Rúbrica

Se evalúan 6 criterios en este examen. Con calificaciones entre 0 y 6 puntos.

Nota: En el caso del menú luego de codificar debe mostrar la codificación asociada a cada caracter.

	Concepto a evaluar	Puntos
1.	Lectura desde archivo	0..2
2.	Implementar la función para determinar las probabilidades o frecuencias	0..2
3.	Función para ordenar las frecuencias y mostrar la tabla	0..2
4.	Estructura de Datos (Arbol Binario) para codificación	0..6
5.	Algoritmo de decodificación	0..4
6.	Menú con opciones, dado una archivo,para codificar y decodificar	0..4

Cuadro 1: Rúbrica del Examen final

3. Anexo

3.1. Encriptado

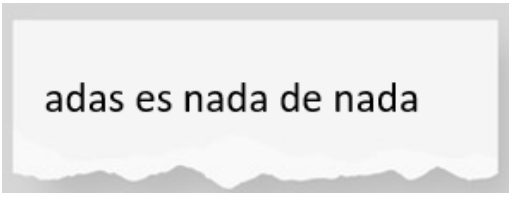

Figura 1: Documento a Encriptar

1. Paso 1: Listado de caracteres y cálculo de ocurrencia

Caracter	Ocurrencia
'a'	6
'd'	4
's'	2
' '	4
'e'	2
'n'	2

2. Paso 2: Calculo de probabilidad y ordenamiento decreciente de la probabilidad. Note que entre el caracter ' ' y el caracter 'd' se da prioridad al ' ' por tener un código ASCII inferior.

Caracter	Probabilidad
'a'	0.30
' '	0.20
'd'	0.20
'e'	0.10
'n'	0.10
's'	0.10

Caracter	Probabilidad	Código
'a'	0.30	
' '	0.20	
'd'	0.20	
'e'	0.10	
'n'	0.10	
's'	0.10	

3. Paso 3: Asociación de Código
4. Pasos 4 al 6: División de listas de forma equilibrada para codificar y construir del árbol de codificación.

   - **Primera iteración**: Observe que conseguimos partir la lista en dos sublistas (['a',' '] y ['d','e','n','s']) con las mismas probabilidades (50 % en la lista superior y 50 % en la lista inferior).

Caracter	Probabilidad	Código
'a'	0.30	0
' '	0.20	0
'd'	0.20	1
'e'	0.10	1
'n'	0.10	1
's'	0.10	1

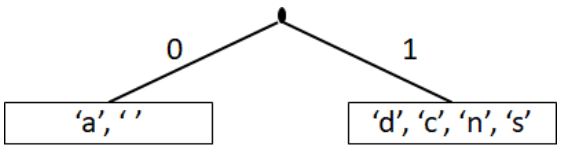

Figura 2: Árbol binario luego de la primera iteración

   - **Segunda iteración**: En este caso la sublista superior (['a',' ']), es vista como la nueva lista a dividir. Observe que no podemos dividirla equilibradamente, aún así procedemos a dividir la lista.

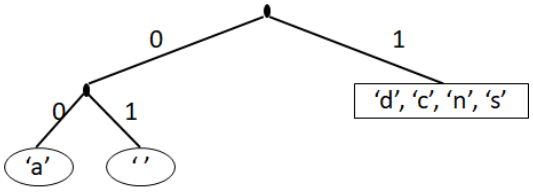

Figura 3: Árbol binario luego de la segunda iteración

Caracter	Probabilidad	Código
'a'	0.30	0 0
' '	0.20	0 1
'd'	0.20	1
'e'	0.10	1
'n'	0.10	1
's'	0.10	1

- **Tercera iteración**: Visto que terminamos de subdividir la sublista superior de la primera iteración, comenzamos a dividir la sublista inferior. Hay diveras formas de dividir esta sublista podriamos separar los tres primeros elementos (['d','e','n']) del último (['s']), pero obtendríamos un gran desequilibrio. Las forma mas equilibrada de separar los elementos serían: separar el primer elemento (['d']) de los últimos (['e','n','s']), o separar los dos primeros elementos (['d', 'e']) de los últimos (['n', 's']). Optaremos por la última opción, pues ante el menor desequilibrio ponderamos el peso de la primera sublista.

Caracter	Probabilidad	Código
'a'	0.30	0 0
' '	0.20	0 1
'd'	0.20	1 0
'e'	0.10	1 0
'n'	0.10	1 1
's'	0.10	1 1

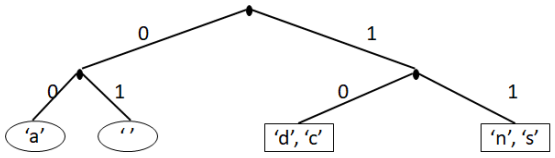

Figura 4: Árbol binario luego de la tercera iteración

- **Cuarta iteración**

Caracter	Probabilidad	Código
'a'	0.30	0 0
' '	0.20	0 1
'd'	0.20	1 0 0
'e'	0.10	1 0 1
'n'	0.10	1 1
's'	0.10	1 1

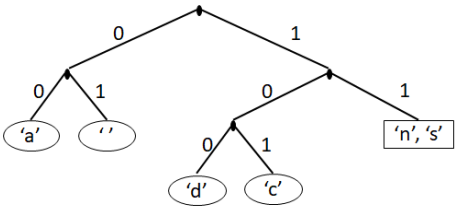

Figura 5: Árbol binario luego de la cuarta iteración

- **Quinta iteración**: Arbol y tabla de codificación generados en el proceso de encriptación.

Caracter	Probabilidad	Código
'a'	0.30	0 0
' '	0.20	0 1
'd'	0.20	1 0 0
'e'	0.10	1 0 1
'n'	0.10	1 1 0
's'	0.10	1 1 1

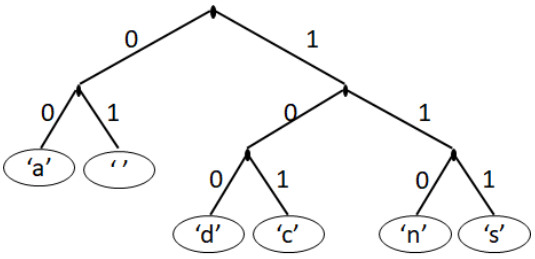

Figura 6: Árbol binario luego de la quinta iteración

3.2. Desencriptado

Dado un documento con una secuencia de 0's y/o 1's, como se muestra en la Figura 7, decodificar el documento usando el árbol binario armado.

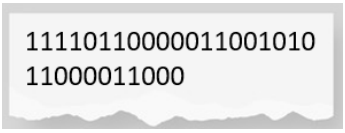

Figura 7: Archivo codificado

La decodificación usando el árbol binario nos mostraría el siguiente mensaje:

”seda de dana”

$$\underbrace{111}_{'s'}\underbrace{101}_{'e'}\underbrace{100}_{'d'}\underbrace{00}_{'a'}\underbrace{01}_{''}\underbrace{100}_{'d'}\underbrace{101}_{'e'}\underbrace{01}_{''}\underbrace{100}_{'d'}\underbrace{00}_{'a'}\underbrace{110}_{'n'}\underbrace{00}_{'a'}$$