



ESCOLA TÈCNICA SUPERIOR
D'ENGINYERIA
Universitat Rovira i Virgili



Estructura de dades

Pràctica 2. Codificació de Huffman

curs 2020-21

Estudiant: Alexandru Chitenco

Professor/a: Clara Granell

Data de lliurament: 15/06/2021

1. Introducció

El objetivo de esta práctica es codificar un mensajes más largos que 1 carácter de manera que la codificación sea lo menos pesada posible. En sí la práctica consiste en dado un mensaje x que cumpla con los requisitos, codificar el mensaje, decodificarlo, mostrar la tabla de codificación y mostrar la eficiencia del algoritmo que usamos con la longitud del mensaje codificado, coste computacional etc.

2. Solució proposada

El algoritmo de Huffman nos ayuda a solucionar este problema. Este algoritmo consiste en relacionar los caracteres más usados de cierto mensaje con los códigos binarios más cortos creados posibles, generados a partir de un árbol creado por una tabla de caracteres con su frecuencia respectiva.

La creación del árbol sigue los siguientes pasos:

1. desmontar el mensaje y crear una tabla con todos los caracteres usados en el mensaje con su frecuencia, es decir en el mensaje "AABBBECCCC" la tabla generada sería:

A	B	E	C
2	3	1	4

Cada carácter con su frecuencia lo llamaremos nodo y el peso será equivalente a la frecuencia

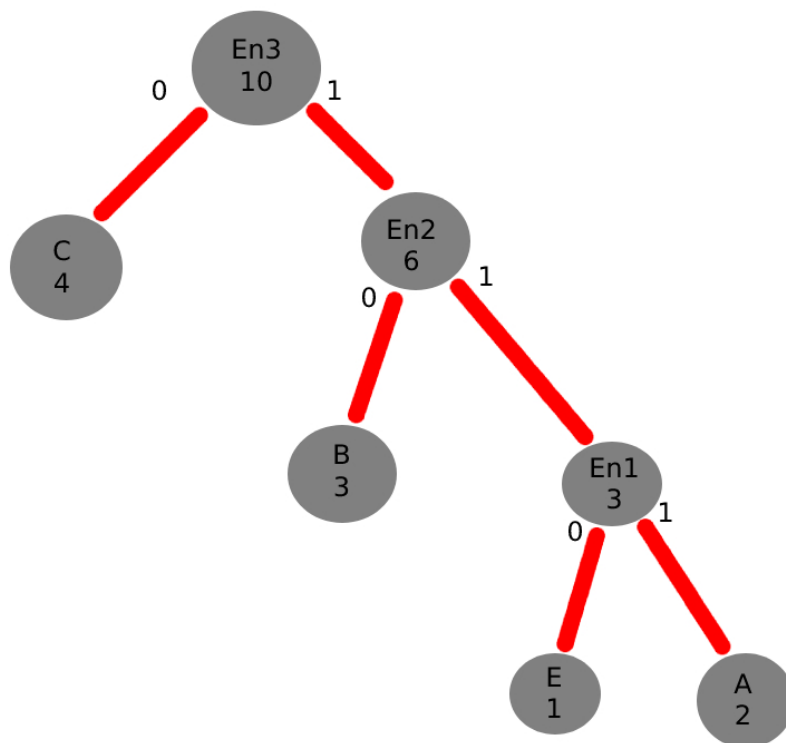
2. Al tener la tabla generada cogemos los nodos de menos peso y los juntamos con un nodo de enlace, el cual su peso equivaldrá a la suma de los pesos de los dos nodos. El enlace tendrá dos punteros que apuntan a los nodos usados. La pareja de nodos se extrae de la tabla y en su lugar se introduce el nodo enlace:

en1(A+E)	B	C
3	3	4

3. repetimos el proceso hasta quedarnos con un único enlace:

en2	C	---	en3
6	4	---	10

Este proceso generaría este árbol:



Y la tabla de codificación se generaría a partir del árbol, siendo equivalente en este ejemplo a:

C	0
B	10
A	111
E	110

3. Descripció dels requeriments

Los requisitos de la práctica son:

1. Dado un mensaje crear su árbol
2. A partir del árbol crear la tabla de codificación
3. ser capaz de codificar un mensaje con la tabla de codificación
4. ser capaz de decodificar un mensaje codificado a partir del árbol
5. medir la eficiencia de nuestra implementación del algoritmo

4. Decisions de disseny

Para hacer la práctica se han tenido que tomar varias decisiones de diseño entre las cuales las más importantes son:

1. Los mensajes a partir de los cuales se genera el árbol no pueden ser inferiores a 1 ni tener un único carácter, en el caso de que el usuario introduzca un mensaje que no cumpla con los requisitos saltará una RuntimeException.
2. Los nodos tienen de atributos dos punteros hacia nodos hijos, un puntero a nodo padre, un int con la frecuencia, un char con la letra usada, y un boolean que se usa en la creación de la tabla de codificación.
3. El árbol se genera a partir de una array de nodos ORDENADOS
4. La tabla de codificación se genera con un puntero de interés PDI que recorre hasta llegar a un final el árbol, al llegar al final apunta ese carácter con el código generado indica que ese nodo ya está apuntado va hacia el nodo padre de nuevo y comprueba si existe algún valor en el nodo hijo siguiente en el caso que están los dos apuntados el padre de marca como apuntado y se eleva un nivel más.

Joc de proves

Para hacer el juego de pruebas usaremos diferentes mensajes(adecuados y no adecuados) para ver el rendimiento de la práctica.

Evaluación de palabras adecuadas:

	Resultado esperado	Resultado obtenido	Resultado esperado	Resultado obtenido	Pasa la prueba
Mensaje	MESSAGE	MESSAGE	asdf_s dedf	asdf_s dedf	SI
Codificación	0101011110 110010	01010111101 10010	01011010111 01111000010 00110111	01011010111 01111000010 00110111	SI
Mida Cod.	2 byte	2 byte	4 byte	4 byte	SI
Mida Arbol	117 byte	117 byte	169 byte	169 byte	SI
Coste Cod.	25	25	54	54	SI
Coste Desco.	23	23	41	41	SI
Coste Tabla	36	36	58	58	SI
Coste Arbol	86	86	197	197	SI
Descodificación	MESSAGE	MESSAGE	asdf_s dedf	asdf_s dedf	SI
Tabla	[<M> <010>] [<A> <011>] [<G> <00>] [<E> <10>] [<S> <11>]	[<M> <010>] [<A> <011>] [<G> <00>] [<E> <10>] [<S> <11>]	[<a> <010>] [<_> <011>] [< > <000>] [<e> <001>] [<s> <110>] [<f> <111>] [<d> <10>]	[<a> <010>] [<_> <011>] [< > <000>] [<e> <001>] [<s> <110>] [<f> <111>] [<d> <10>]	SI

```
"C:\Program Files\Java\jdk-14.0.1\bin\java.exe" "-javaagent:D:\IntelliJ IDEA 2020.3.2\lib\idea_rt.jar=50237:D:\IntelliJ IDEA 2020.3.2\bin" -
escriba la secuencia que desea, que sea mas larga de 1 caracter y utilizar al menos dos caracteres distintos
MESSAGE
[<M> <010>]
[<A> <011>]
[<G> <00>]
[<E> <10>]
[<S> <11>]

010101110110010 mida mensaje:2byte Coste computacional codificacion: 25
mida arbol:9 nodos(char 2byte, int 4byte puntero 2*3byte, boolean 1byte = 13byte) que equivalen a 117 bytes Coste computacional arbol: 86
El codigo descodificado: MESSAGE Coste computacional descodificacion: 23
El coste computacional de la creacion de la tabla ha sido: 36

Process finished with exit code 0
|
```

```

C:\Program Files\Java\jdk-14.0.1\bin\java.exe -javaagent:D:\IntelliJ IDEA 2020.3.2\lib\idea_rt.jar=50705:D:\IntelliJ IDEA 2020.3.2\bin
escriba la secuencia que desea, que sea mas larga de 1 caracter y utilizar al menos dos caracteres distintos
asdf_s dedf
[<a> <010>]
[<_> <011>]
[< > <000>]
[<e> <001>]
[<s> <110>]
[<f> <111>]
[<d> <10>]

01011010111011100001000110111 mida mensaje:4byte Coste computacional codificacion: 54
mida arbol:13 nodos(char 2byte, int 4byte puntero 2*3byte, boolean 1byte = 13byte) que equivalen a 169 bytes Coste computacional arbol: 197
El codigo descodificado: asdf_s dedf Coste computacional descodificacion: 41
El coste computacional de la creacion de la tabla ha sido: 58

Process finished with exit code 0

```

Limitacions i punts no avaluats (si s'escau)

Para mensajes que no cumplen con los requisitos sale un RuntimeException:

```

C:\Program Files\Java\jdk-14.0.1\bin\java.exe -javaagent:D:\IntelliJ IDEA 2020.3.2\lib\idea_rt.jar=50074:D\
escriba la secuencia que desea, que sea mas larga de 1 caracter y utilizar al menos dos caracteres distintos
11111111111111111111
Exception in thread "main" java.lang.RuntimeException:
tienes que usar mas de un caracter
    at Main.crearArbol(Main.java:62)
    at Main.main(Main.java:15)

Process finished with exit code 1
|

```

```

C:\Program Files\Java\jdk-14.0.1\bin\java.exe -javaagent:D:\IntelliJ IDEA 2020.3.2\lib\idea_rt.jar=55507:D\
escriba la secuencia que desea, que sea mas larga de 1 caracter y utilizar al menos dos caracteres distintos
0
Exception in thread "main" java.lang.RuntimeException:
medida de mensaje inadecuada
    at Main.main(Main.java:13)

Process finished with exit code 1
|

```

Conclusions

Esta práctica me ha parecido más interesante, menos tediosa y más sencilla que la primera, tuve algunos problemas para implementar los algún método de la clase Huffman(sobretudo en la creación del árbol y la tabla de codificación) pero en general la hice en un periodo corto de tiempo (aunque fuera de primera convocatoria desgraciadamente).

Observaciones

Esta la gestión de versiones en mi GitHub -> <https://github.com/liteJoseStalin/Huffman>

La práctica esta hecha con IntelliJ