



Textos y Cadenas

Codificación y compresión

Comprimir es comprender

Gregory Chaitin

codificación por longitud de series

El tipo más sencillo de redundancia que podemos encontrar en una cadena son las largas series de caracteres repetidos

AAAABBBBAABBBBBBBBBBBBBCCCCCCCCDABAAABBBBCCCD

Este tipo de cadena se puede expresar de forma más compacta reemplazando las repeticiones de un caracter por una sola ocurrencia del caracter precedida por un valor que indique la cantidad de veces que se repite

4A3BAA5B8CDABCB3A4B3CD

Este tipo de codificación se conoce como codificación por longitud de series

inconvenientes

¿Qué sucede si la cadena incluye números?

Podríamos usar un caracter de escape

¿Qué sucede si el carácter de escape es también parte del mensaje?

codificación de longitud fija

Longitud = 40

AAAEABBBAAABBBEEBBCCCCCCCCDABAEAAABBBBCCCD

Asumiendo que está codificada en ASCII la longitud de la cadena es $40 \times 8 \text{ bits} = \mathbf{320 \text{ bits}}$

codificación de longitud fija

Longitud = 40

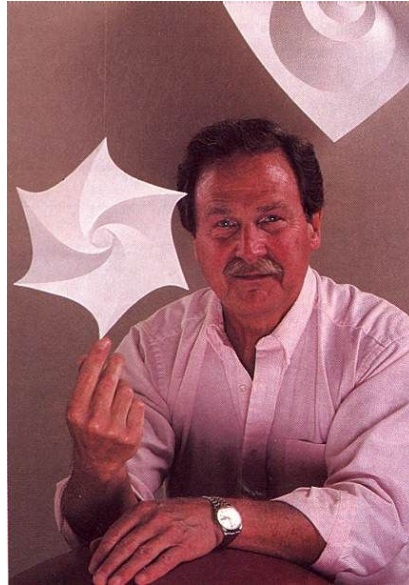
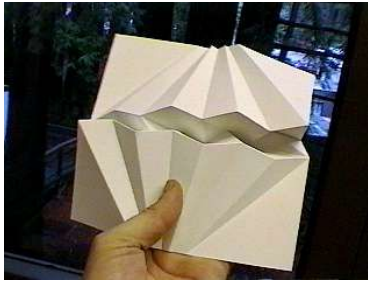
AAAEABBBAAABBBEEBBCCCCCCCCDABAEAAABBBBCCCD

El alfabeto del mensaje tiene solo 5 símbolos
Es decir que los puedo representar con solo 3 bits ya que $6 < 2^3$

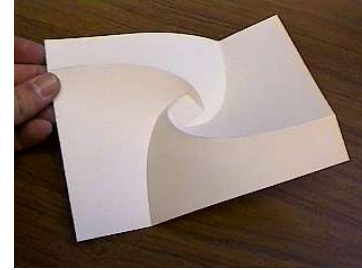
char	cod
A	001
B	010
C	011
D	100
E	101

Con esta nueva codificación necesito $40 \times 3 \text{ bits} = 120 \text{ bits}$
Más la tabla que tiene $5 \times 8 \text{ bits} + 5 \times 3 \text{ bits} = 55 \text{ bits}$
Total 175 bits

codificación de longitud variable: Huffman



David A. Huffman



codificación de longitud variable: Huffman

Longitud = 40

AAAEABBBBAABBBEEBBCCCCCCCCDABAEAAABBBBCCCD

char	cuenta	cod
A	10	
B	13	
C	11	
D	2	
E	4	

2

D

4

E

10

A

11

C

13

B

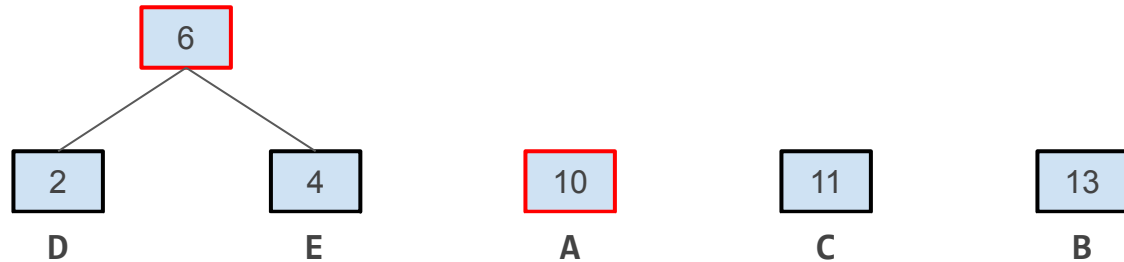
en orden creciente por frecuencia

codificación de longitud variable: Huffman

Longitud = 40

AAAEABBBBAABBBEEBBCCCCCCCCDABAEAAABBBBCCCD

char	cuenta	cod
A	10	
B	13	
C	11	
D	2	
E	4	



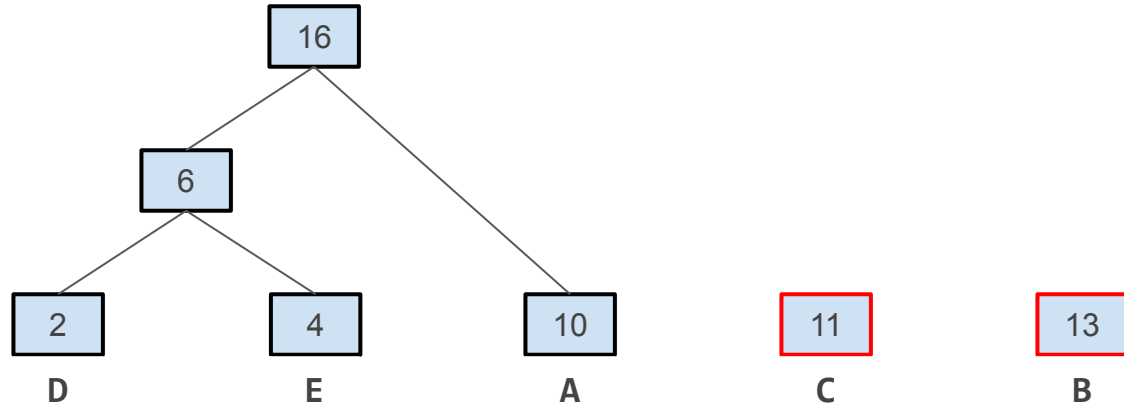
en orden creciente por frecuencia →

codificación de longitud variable: Huffman

Longitud = 40

AAAEABBBAAABBBEEBBCCCCCCCCDABAEAAABBBBCCCD

char	cuenta	cod
A	10	
B	13	
C	11	
D	2	
E	4	



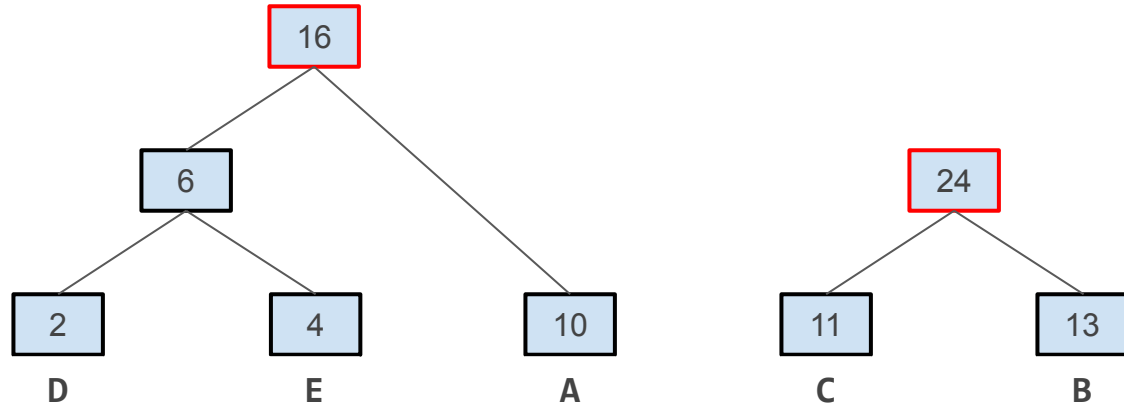
en orden creciente por frecuencia →

codificación de longitud variable: Huffman

Longitud = 40

AAAEABBBBAABBBEEBBCCCCCCCCDABAEAAABBBBCCCD

char	cuenta	cod
A	10	
B	13	
C	11	
D	2	
E	4	



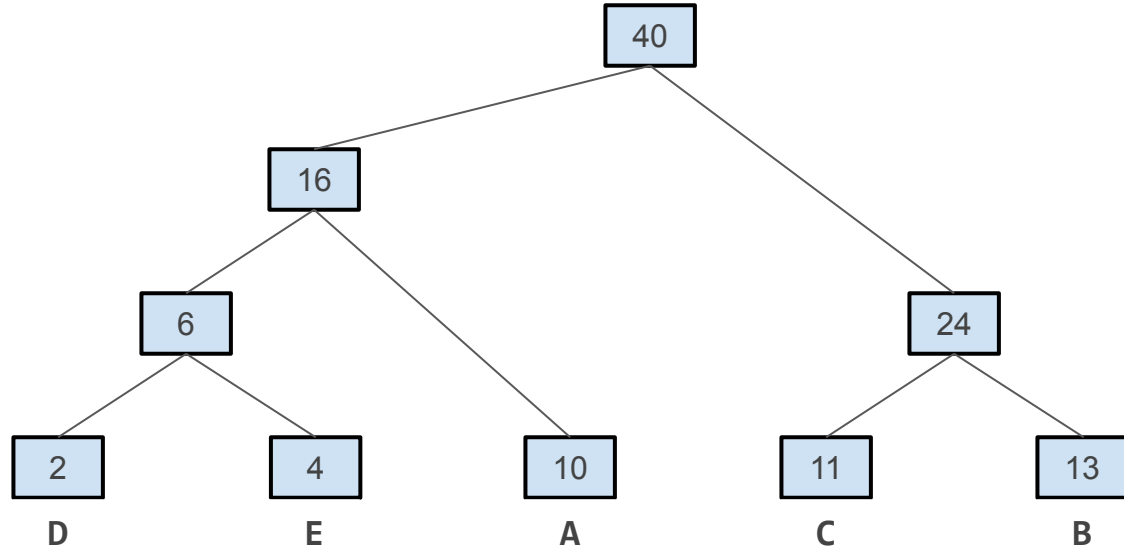
en orden creciente por frecuencia →

codificación de longitud variable: Huffman

Longitud = 40

AAAEABBBBAABBBEEBBCCCCCCCCDABAEAAABBBBCCCD

char	cuenta	cod
A	10	
B	13	
C	11	
D	2	
E	4	



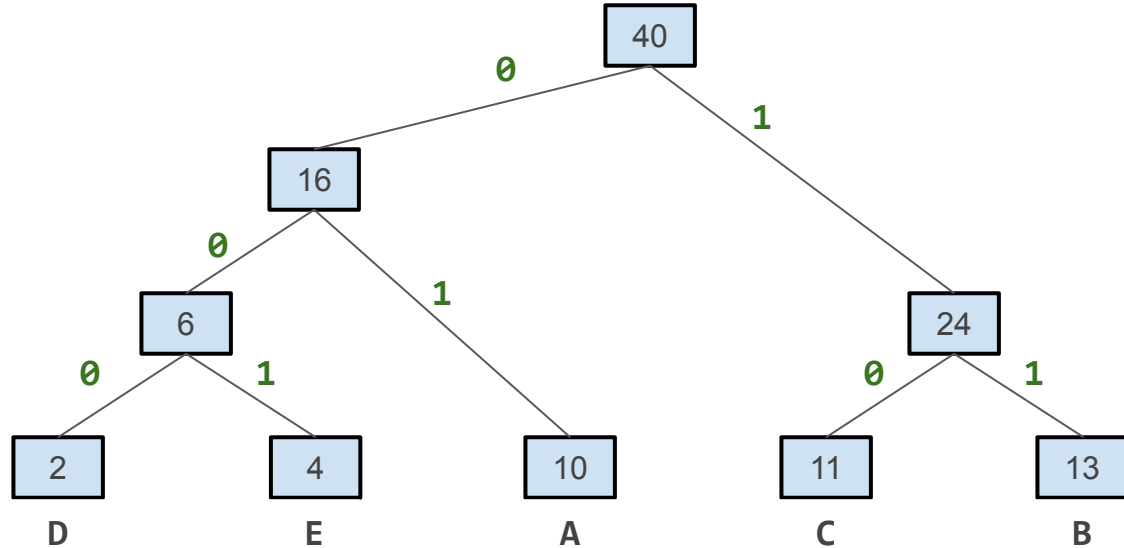
en orden creciente por frecuencia →

codificación de longitud variable: Huffman

Longitud = 40

AAAEABBBBAABBBEEBBCCCCCCCCDABAEAAABBBBCCCD

char	cuenta	cod
A	10	
B	13	
C	11	
D	2	
E	4	



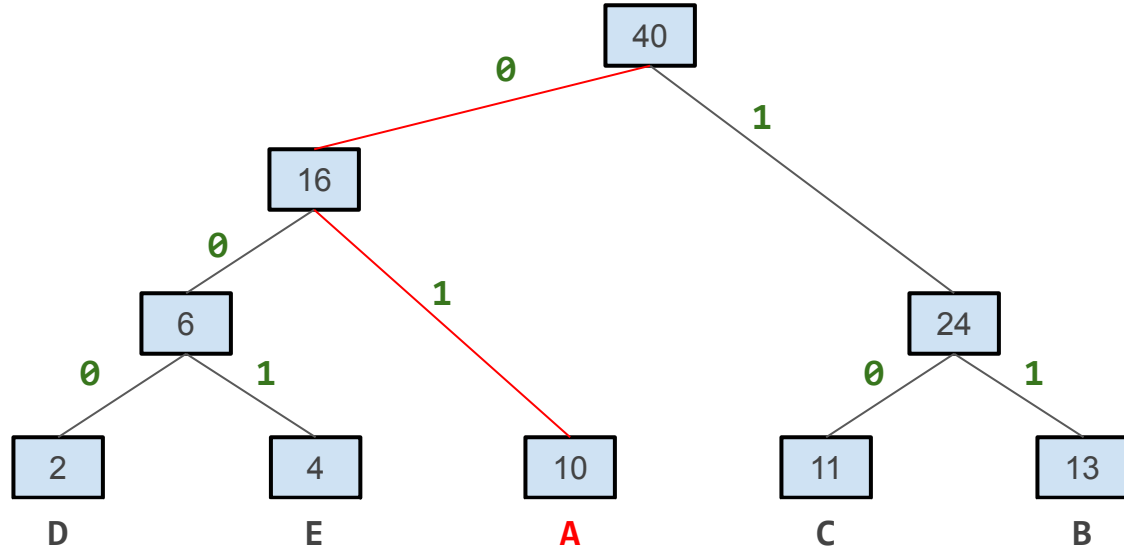
en orden creciente por frecuencia →

codificación de longitud variable: Huffman

Longitud = 40

AAAEABBBBAABBBEEBBCCCCCCCCDABAEAAABBBBCCCD

char	cuenta	cod
A	10	01
B	13	
C	11	
D	2	
E	4	



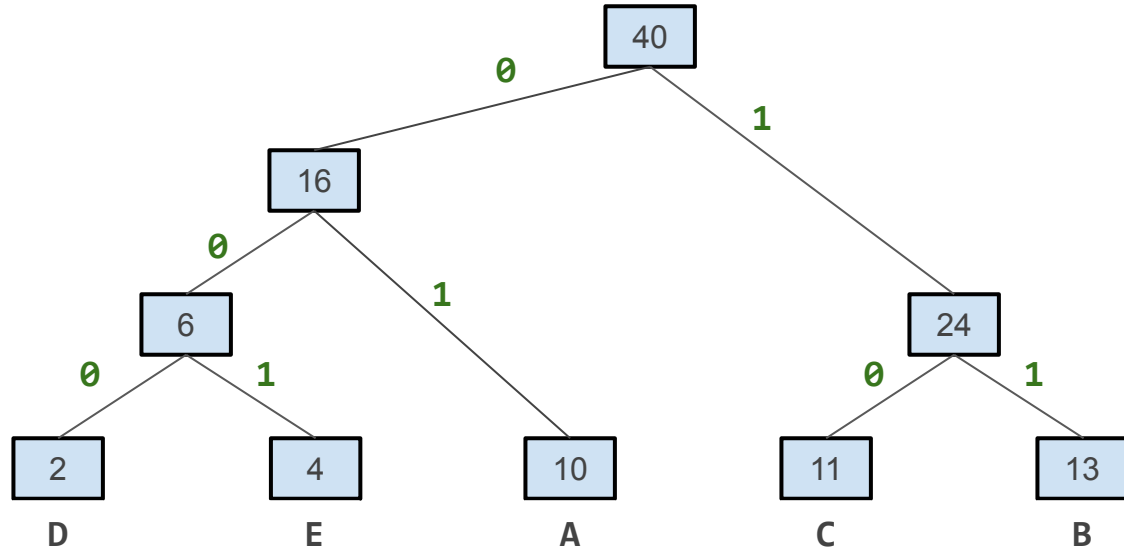
en orden creciente por frecuencia →

codificación de longitud variable: Huffman

Longitud = 40

AAAEABBBBAABBBEEBBCCCCCCCCDABAEAAABBBBCCCD

char	cuenta	cod
A	10	01
B	13	11
C	11	10
D	2	000
E	4	001



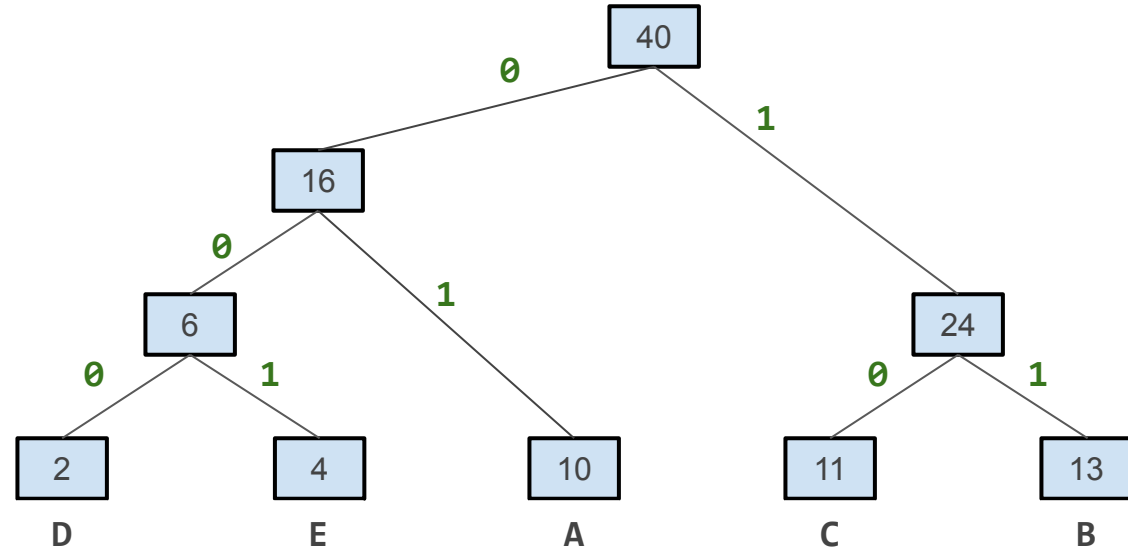
en orden creciente por frecuencia →

codificación de longitud variable: Huffman

Longitud = 40

AAAEABBBBAABBBEEBBCCCCCCCCDABAEABBBBCCCD

char	cuenta	cod	tamaño
A	10	01	20
B	13	11	26
C	11	10	22
D	2	000	6
E	4	001	12
			86
			Tabla 52
			138



en orden creciente por frecuencia →



Ejercicio 5.05.

Implementá el algoritmo para obtener el código de Huffman de una cadena dada



Ejercicio 5.06.

Hallá la codificación de Huffman para la siguiente cadena

abffdfffffffddfffcffddefeffffffeefffedcdffffdfdeffffff
fefeecfbfaffffdddefeffffffeeeffdffcdafeebcacbbbacffbc
cbcbcc

a	b	c	d	e	f
5	9	12	13	16	45



Ejercicio 5.07.

Escribir un programa que reemplace los caracteres duplicados en una cadena por el caracter `_` (`ascii 95`)

`universidad blas pascal`

`univers_da_ bl__ p__c__`



Ejercicio 5.08.

Escribir un programa que encuentre un patrón en un texto, con la siguiente variante: el caracter `*` (`ascii 42`) en el patrón equivale a un comodín y puede coincidir con cualquier letra en esa posición

texto: la **masa** sobre la **mesa**

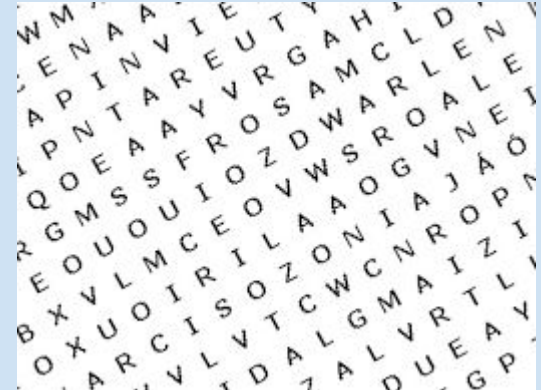
patrón: `m*sa`



Ejercicio 5.09.

Escribir un programa que encuentre un patrón en una sopa de letras y decir si se encontró en dirección horizontal, vertical, diagonal izquierda o diagonal derecha

```
mat[N][N] = { {'a', 'v', 't', 'h', 'e'},  
               {'f', 'm', 'w', 'm', 'j'},  
               {'k', 'l', 'e', 'e', 'o'},  
               {'m', 'q', 'r', 's', 't'},  
               {'u', 'c', 'k', 'a', 'a'}};  
  
patron = "mesa";
```





Ejercicio 5.10.

Escribí un programa que encuentre las cadenas palíndromas de 6, 5, 4 ó 3 caracteres en un texto

adgt**add**agtad**bbb**gaattbaadt**att**ab