

02_Huffman_Laboratorio

March 7, 2022

```
[20]: import time
import random
import itertools
```

Definir una función `LongitudMedia(C,p)` que dado un código C y una distribución de probabilidades p devuelva la longitud media \bar{l} del código.

```
[2]: def LongitudMedia(C,p):

    return 1
```

Definir una función `Huffman(p)` que dada una distribución de probabilidades p devuelva UN código de Huffman asociado a dicha la distribución de probabilidades.

```
[80]: def Huffman(p):

    return CodeCanónico(longitudes_palabras)
```

0.0.1 Ejemplos

```
[17]: p=[0.5,0.1,0.2,0.1,0.05,0.05]
codigo=Huffman(p)
print(p)
print([[p[i],codigo[i]] for i in range(len(p))], '\n\nLongitud Media:
→',LongitudMedia(codigo,p))
```

```
[0.5, 0.1, 0.2, 0.1, 0.05, 0.05]
[[0.5, '0'], [0.1, '100'], [0.2, '101'], [0.1, '110'], [0.05, '1110'], [0.05,
'1111']]
```

Longitud Media: 2.1

```
[43]: p=[0.8,0.1,0.05,0.05]
codigo=Huffman(p)
print(p)
print([[p[i],codigo[i]] for i in range(len(p))], '\n Longitud Media:',
→LongitudMedia(codigo,p))
```

```
[0.8, 0.1, 0.05, 0.05]
[[0.8, '0'], [0.1, '10'], [0.05, '110'], [0.05, '111']]
Longitud Media: 1.2999999999999998
```

```
[44]: p=[0.5,0.25,0.1,0.05,0.05,0.05]
codigo=Huffman(p)
print(p)
print([[p[i],codigo[i]] for i in range(len(p))], '\n Longitud Media:',
      ↳LongitudMedia(codigo,p))
```

```
[0.5, 0.25, 0.1, 0.05, 0.05, 0.05] 1.0
[[0.5, '0'], [0.25, '10'], [0.1, '110'], [0.05, '11110'], [0.05, '1110'], [0.05,
'11111']]
Longitud Media: 2.0
```

```
[45]: p=[0.102, 0.106, 0.053, 0.114, 0.0081, 0.106, 0.0088, 0.030, 0.056, 0.055, 0.
      ↳032, 0.094, 0.0075, 0.078, 0.1496]
codigo=Huffman(p)
print(p)
print([[p[i],codigo[i]] for i in range(len(p))], '\n Longitud Media:',
      ↳LongitudMedia(codigo,p))
```

```
[0.102, 0.106, 0.053, 0.114, 0.0081, 0.106, 0.0088, 0.03, 0.056, 0.055, 0.032,
0.094, 0.0075, 0.078, 0.1496] 1.0
[[0.102, '000'], [0.106, '001'], [0.053, '11100'], [0.114, '010'], [0.0081,
'1111110'], [0.106, '011'], [0.0088, '111110'], [0.03, '11101'], [0.056,
'1010'], [0.055, '1011'], [0.032, '11110'], [0.094, '1100'], [0.0075,
'1111111'], [0.078, '1101'], [0.1496, '100']]
Longitud Media: 3.6018000000000003
```

```
[52]: n=2**7
p=[1/n for _ in range(n)]

codigo=Huffman(p)

print(codigo, '\n Longitud Media:', LongitudMedia(codigo,p))
```

```
['0000000', '0000001', '0000010', '0000011', '0000100', '0000101', '0000110',
'0000111', '0001000', '0001001', '0001010', '0001011', '0001100', '0001101',
'0001110', '0001111', '0010000', '0010001', '0010010', '0010011', '0010100',
'0010101', '0010110', '0010111', '0011000', '0011001', '0011010', '0011011',
'0011100', '0011101', '0011110', '0011111', '0100000', '0100001', '0100010',
'0100011', '0100100', '0100101', '0100110', '0100111', '0101000', '0101001',
'0101010', '0101011', '0101100', '0101101', '0101110', '0101111', '0110000',
'0110001', '0110010', '0110011', '0110100', '0110101', '0110110', '0110111',
'0111000', '0111001', '0111010', '0111011', '0111100', '0111101', '0111110',
```

```
'0111111', '1000000', '1000001', '1000010', '1000011', '1000100', '1000101',
'1000110', '1000111', '1001000', '1001001', '1001010', '1001011', '1001100',
'1001101', '1001110', '1001111', '1010000', '1010001', '1010010', '1010011',
'1010100', '1010101', '1010110', '1010111', '1011000', '1011001', '1011010',
'1011011', '1011100', '1011101', '1011110', '1011111', '1100000', '1100001',
'1100010', '1100011', '1100100', '1100101', '1100110', '1100111', '1101000',
'1101001', '1101010', '1101011', '1101100', '1101101', '1101110', '1101111',
'1110000', '1110001', '1110010', '1110011', '1110100', '1110101', '1110110',
'1110111', '1111000', '1111001', '1111010', '1111011', '1111100', '1111101',
'1111110', '1111111']
```

Longitud Media: 7.0

Definir una función `tablaFrecuencias(mensaje, numero_de_simbolos=1)` que dado un mensaje M devuelva una tabla con la frecuencia de cada tupla de `numero_de_simbolos` símbolos del mensaje.

```
[55]: def tablaFrecuencias(mensaje, numero_de_simbolos=1):
      return frecuencias
```

0.0.2 Ejemplos de como cambian las frecuencias según la fuente S

```
[58]:
```

```

mensaje='La heroica ciudad dormía la siesta. El viento Sur, caliente y
→perezoso, empujaba las nubes blanquecinas que se rasgaban al correr hacia el
→Norte. En las calles no había más ruido que el rumor estridente de los
→remolinos de polvo, trapos, pajas y papeles que iban de arroyo en arroyo, de
→acera en acera, de esquina en esquina revolando y persiguiéndose, como
→mariposas que se buscan y huyen y que el aire envuelve en sus pliegues
→invisibles. Cual turbas de pilluelos, aquellas migajas de la basura,
→aquellas sobras de todo se juntaban en un montón, parábanse como dormidas un
→momento y brincaban de nuevo sobresaltadas, dispersándose, trepando unas por
→las paredes hasta los cristales temblorosos de los faroles, otras hasta los
→carteles de papel mal pegado a las esquinas, y había pluma que llegaba a un
→tercer piso, y arenilla que se incrustaba para días, o para años, en la
→vidriera de un escaparate, agarrada a un plomo. Vetusta, la muy noble y leal
→ciudad, corte en lejano siglo, hacía la digestión del cocido y de la olla
→podrida, y descansaba oyendo entre sueños el monótono y familiar zumbido de
→la campana de coro, que retumbaba allá en lo alto de la esbeltatorre en la
→Santa Basílica. La torre de la catedral, poema romántico de piedra,delicado
→himno, de dulces líneas de belleza muda y perenne, era obra del siglo diez y
→seis, aunque antes comenzada, de estilo gótico, pero, cabe decir, moderado
→por un instinto de prudencia y armonía que modificaba las vulgares
→exageraciones de esta arquitectura. La vista no se fatigaba contemplando
→horas y horas aquel índice depiedra que señalaba al cielo; no era una de
→esas torres cuya aguja se quiebra desutil, más flacas que esbeltas,
→amaneradas, como señoritas cursis que aprietandemasiado el corsé; era maciza
→sin perder nada de su espiritual grandeza, y hasta sussegundos corredores,
→elegante balaustrada, subía como fuerte castillo, lanzándosedesde allí en
→pirámide de ángulo gracioso, inimitable en sus medidas y proporciones.Como
→haz de músculos y nervios la piedra enroscándose en la piedra trepaba a la
→altura, haciendo equilibrios de acróbata en el aire; y como prodigio de
→juegosmalabares, en una punta de caliza se mantenía, cual imantada, una bola
→grande debronce dorado, y encima otra más pequeña, y sobre ésta una cruz de
→hierro que acababa en pararrayos.'
```

```

nnn=1
print(tablaFrecuencias(mensaje,numero_de_simbolos=nnn))
```

```

[[' ', 384], [',', 46], ['.', 8], [';', 3], ['B', 1], ['C', 2], ['E', 2], ['L',
3], ['N', 1], ['S', 2], ['V', 1], ['a', 267], ['b', 46], ['c', 64], ['d', 104],
['e', 238], ['f', 6], ['g', 23], ['h', 15], ['i', 98], ['j', 7], ['l', 112],
['m', 49], ['n', 110], ['o', 141], ['p', 48], ['q', 25], ['r', 126], ['s', 145],
['t', 68], ['u', 87], ['v', 11], ['x', 1], ['y', 29], ['z', 11], ['á', 11],
['é', 3], ['í', 12], ['ñ', 5], ['ó', 5], ['ú', 1]]
```

[62]:

```

mensaje='La heroica ciudad dormía la siesta. El viento Sur, caliente y
→perezoso, empujaba las nubes blanquecinas que se rasgaban al correr hacia el
→Norte. En las calles no había más ruido que el rumor estridente de los
→remolinos de polvo, trapos, pajas y papeles que iban de arroyo en arroyo, de
→acera en acera, de esquina en esquina revolando y persiguiéndose, como
→mariposas que se buscan y huyen y que el aire envuelve en sus pliegues
→invisibles. Cual turbas de pilluelos, aquellas migajas de la basura,
→aquellas sobras de todo se juntaban en un montón, parábanse como dormidas un
→momento y brincaban de nuevo sobresaltadas, dispersándose, trepando unas por
→las paredes hasta los cristales temblorosos de los faroles, otras hasta los
→carteles de papel mal pegado a las esquinas, y había pluma que llegaba a un
→tercer piso, y arenilla que se incrustaba para días, o para años, en la
→vidriera de un escaparate, agarrada a un plomo. Vetusta, la muy noble y leal
→ciudad, corte en lejano siglo, hacía la digestión del cocido y de la olla
→podrida, y descansaba oyendo entre sueños el monótono y familiar zumbido de
→la campana de coro, que retumbaba allá en lo alto de la esbeltatorre en la
→Santa Basílica. La torre de la catedral, poema romántico de piedra,delicado
→himno, de dulces líneas de belleza muda y perenne, era obra del siglo diez y
→seis, aunque antes comenzada, de estilo gótico, pero, cabe decir, moderado
→por un instinto de prudencia y armonía que modificaba las vulgares
→exageraciones de esta arquitectura. La vista no se fatigaba contemplando
→horas y horas aquel índice depiedra que señalaba al cielo; no era una de
→esas torres cuya aguja se quiebra desutil, más flacas que esbeltas,
→amaneradas, como señoritas cursis que aprietandemasiado el corsé; era maciza
→sin perder nada de su espiritual grandeza, y hasta sussegundos corredores,
→elegante balaustrada, subía como fuerte castillo, lanzándosedesde allí en
→pirámide de ángulo gracioso, inimitable en sus medidas y proporciones.Como
→haz de músculos y nervios la piedra enroscándose en la piedra trepaba a la
→altura, haciendo equilibrios de acróbata en el aire; y como prodigio de
→juegosmalabares, en una punta de caliza se mantenía, cual imantada, una bola
→grande debronce dorado, y encima otra más pequeña, y sobre ésta una cruz de
→hierro que acababa en pararrayos.'
```

```
nnn=2
```

```
print(tablaFrecuencias(mensaje,numero_de_simbolos=nnn))
```

```

[[' C', 1], [' L', 1], [' N', 1], [' a', 18], [' b', 5], [' c', 17], [' d', 23],
[' e', 21], [' f', 2], [' g', 2], [' h', 5], [' i', 4], [' j', 2], [' l', 15],
[' m', 7], [' n', 3], [' o', 4], [' p', 15], [' q', 8], [' r', 5], [' s', 9], ['
t', 2], [' u', 5], [' v', 1], [' y', 9], [' á', 1], [' é', 1], [' ', 16],
[' ,d', 1], [' . ', 4], [' . . ', 1], [' .C', 1], [' ; ', 1], [' Ba', 1], [' El', 1],
[' En', 1], [' La', 2], [' Sa', 1], [' Su', 1], [' Ve', 1], [' a ', 40], [' a,', 8],
[' a.', 1], [' ab', 9], [' ac', 5], [' ad', 7], [' ag', 1], [' ai', 1], [' al', 9],
[' am', 1], [' an', 9], [' ap', 2], [' aq', 2], [' ar', 7], [' as', 10], [' at', 3],
[' ay', 1], [' añ', 1], [' ba', 14], [' bl', 1], [' br', 3], [' ca', 7], [' ce', 2],
[' ci', 7], [' co', 5], [' cr', 3], [' ct', 1], [' cu', 3], [' cí', 1], [' d ', 1],
[' d,', 1], [' da', 7], [' de', 23], [' di', 4], [' do', 11], [' dr', 4], [' du', 1],
```

```

['e ', 34], ['e,', 3], ['ea', 1], ['eb', 2], ['ed', 3], ['ei', 1], ['el', 11],
['em', 2], ['en', 15], ['ep', 3], ['eq', 1], ['er', 10], ['es', 20], ['et', 1],
['ev', 2], ['ez', 2], ['eñ', 2], ['fa', 1], ['fl', 1], ['fu', 1], ['ga', 7],
['ge', 1], ['gi', 1], ['gl', 1], ['go', 1], ['gr', 2], ['gu', 3], ['ha', 5],
['hi', 2], ['ho', 2], ['hu', 1], ['ia', 3], ['ib', 2], ['ic', 3], ['id', 4],
['ie', 9], ['if', 1], ['ig', 2], ['il', 2], ['im', 1], ['in', 2], ['io', 3],
['ir', 2], ['is', 5], ['it', 1], ['iu', 2], ['iz', 1], ['ié', 1], ['ja', 4], ['l',
', 10], ['l,', 1], ['la', 14], ['lc', 1], ['le', 10], ['lg', 1], ['li', 4],
['ll', 6], ['lo', 9], ['lt', 1], ['lu', 1], ['ma', 8], ['mb', 2], ['me', 2],
['mi', 2], ['mn', 1], ['mo', 7], ['mp', 2], ['mu', 1], ['má', 2], ['n ', 14],
['na', 6], ['nc', 4], ['nd', 8], ['ne', 2], ['ng', 1], ['ni', 1], ['nn', 1],
['no', 6], ['nq', 1], ['ns', 2], ['nt', 8], ['nu', 1], ['nv', 1], ['nz', 1],
['nó', 1], ['o ', 20], ['o,', 6], ['o;', 1], ['ob', 2], ['oc', 1], ['od', 4],
['oe', 1], ['oi', 1], ['ol', 5], ['om', 6], ['on', 4], ['op', 1], ['or', 8],
['os', 8], ['ot', 1], ['pa', 7], ['pe', 8], ['pi', 3], ['pl', 1], ['po', 3],
['pr', 2], ['pu', 1], ['qu', 10], ['r ', 5], ['r,', 2], ['ra', 18], ['rb', 1],
['rc', 1], ['rd', 1], ['re', 11], ['ri', 5], ['rm', 2], ['ro', 7], ['rq', 1],
['rr', 4], ['rs', 3], ['rt', 1], ['ru', 3], ['rv', 1], ['rá', 1], ['s ', 34],
['s,', 8], ['s.', 1], ['sa', 3], ['sb', 2], ['sc', 3], ['se', 10], ['si', 4],
['sm', 1], ['so', 4], ['sq', 2], ['st', 6], ['su', 4], ['sí', 1], ['ta', 11],
['te', 9], ['ti', 6], ['to', 5], ['tr', 7], ['tu', 3], ['u ', 1], ['ua', 1],
['ub', 2], ['ud', 1], ['ue', 12], ['ui', 6], ['uj', 1], ['ul', 1], ['um', 1],
['un', 9], ['ur', 2], ['us', 3], ['ut', 1], ['uy', 1], ['uz', 1], ['ve', 1],
['vi', 2], ['vo', 1], ['vu', 2], ['xa', 1], ['y ', 14], ['ye', 2], ['yo', 2],
['z ', 1], ['za', 2], ['zo', 1], ['zu', 1], ['zá', 1], ['á ', 1], ['ám', 1],
['án', 3], ['ás', 1], ['é;', 1], ['í ', 1], ['ía', 7], ['ín', 2], ['ña', 1],
['ño', 1], ['ób', 1], ['ón', 2], ['ót', 1], ['ús', 1]]

```

Definir una función `EncodeHuffman(mensaje,numero_de_simbolos=n_simbolos)` **que codifique un mensaje utilizando un código de Huffman obtenido a partir de las frecuencias de los caracteres del mensaje.**

Codifico un mensaje de prueba tomando las frecuencias de cada símbolo.

Hago una estimación de la ratio de compresión sin contar que he de almacenar el diccionario y contándolo.

Los tiempos dependerán del entorno de ejecución.

[66]:

```

mensaje='La heroica ciudad dormía la siesta. El viento Sur, caliente y
→perezoso, empujaba las nubes blanquecinas que se rasgaban al correr hacia el
→Norte. En las calles no había más ruido que el rumor estridente de los
→remolinos de polvo, trapos, pajas y papeles que iban de arroyo en arroyo, de
→acera en acera, de esquina en esquina revolando y persiguiéndose, como
→mariposas que se buscan y huyen y que el aire envuelve en sus pliegues
→invisibles. Cual turbas de pilluelos, aquellas migajas de la basura,
→aquellas sobras de todo se juntaban en un montón, parábanse como dormidas un
→momento y brincaban de nuevo sobresaltadas, dispersándose, trepando unas por
→las paredes hasta los cristales temblorosos de los faroles, otras hasta los
→carteles de papel mal pegado a las esquinas, y había pluma que llegaba a un
→tercer piso, y arenilla que se incrustaba para días, o para años, en la
→vidriera de un escaparate, agarrada a un plomo. Vetusta, la muy noble y leal
→ciudad, corte en lejano siglo, hacía la digestión del cocido y de la olla
→podrida, y descansaba oyendo entre sueños el monótono y familiar zumbido de
→la campana de coro, que retumbaba allá en lo alto de la esbeltatorre en la
→Santa Basílica. La torre de la catedral, poema romántico de piedra,delicado
→himno, de dulces líneas de belleza muda y perenne, era obra del siglo diez y
→seis, aunque antes comenzada, de estilo gótico, pero, cabe decir, moderado
→por un instinto de prudencia y armonía que modificaba las vulgares
→exageraciones de esta arquitectura. La vista no se fatigaba contemplando
→horas y horas aquel índice depiedra que señalaba al cielo; no era una de
→esas torres cuya aguja se quiebra desutil, más flacas que esbeltas,
→amaneradas, como señoritas cursis que aprietandemasiado el corsé; era maciza
→sin perder nada de su espiritual grandeza, y hasta sussegundos corredores,
→elegante balaustrada, subía como fuerte castillo, lanzándosedesde allí en
→pirámide de ángulo gracioso, inimitable en sus medidas y proporciones.Como
→haz de músculos y nervios la piedra enroscándose en la piedra trepaba a la
→altura, haciendo equilibrios de acróbata en el aire; y como prodigio de
→juegosmalabares, en una punta de caliza se mantenía, cual imantada, una bola
→grande debronce dorado, y encima otra más pequeña, y sobre ésta una cruz de
→hierro que acababa en pararrayos.'
```

```
mensaje=mensaje[:]*100
```

```
n_simbolos=1
```

```
mensaje_codificado,
```

```
→m2c,len_M=EncodeHuffman(mensaje,numero_de_simbolos=n_simbolos)
```

```
mensaje_recuperado=DecodeHuffman(mensaje_codificado,m2c,len_M)
```

```
print('\n Diccionario ordenado alfabéticamente: ', sorted(m2c.items(),
→key=lambda x: x[0]))
```

```
print('\n Diccionario ordenado longitud código: ', sorted(m2c.items(),
→key=lambda x: len(x[1])))
```

```
ratio_compresion=8*len(mensaje)/len(mensaje_codificado)
```

```
ratio_compresion_con_diccionario=8*len(mensaje)/
```

```
→(len(mensaje_codificado)+len(m2c)*8*(n_simbolos+1))
```

```
Tiempo para crear tabla de frecuencias: 0.11396598815917969
Entropía por símbolo: 4.201875546880906
Ratio de compresión máxima estimada sin diccionario: 1.9039116962753644
Tiempo para crear código Huffman: 0.000690460205078125
Tiempo para codificar: 0.033026933670043945

Diccionario ordenado alfabéticamente: [(' ', '110'), (',', '111001'), ('.', '10110000'), (';', '000010001'), ('B', '0000000100'), ('C', '1011000101'), ('E', '1011000100'), ('L', '000010000'), ('N', '10110001111'), ('S', '0000000101'), ('V', '10110001110'), ('a', '011'), ('b', '111000'), ('c', '10000'), ('d', '11111'), ('e', '010'), ('f', '00000000'), ('g', '1011011'), ('h', '0000101'), ('i', '11110'), ('j', '00001001'), ('l', '0010'), ('m', '111011'), ('n', '0001'), ('o', '1001'), ('p', '111010'), ('q', '000001'), ('r', '0011'), ('s', '1010'), ('t', '10001'), ('u', '10111'), ('v', '10110101'), ('x', '10110001101'), ('y', '000011'), ('z', '10110100'), ('á', '10110011'), ('é', '000000011'), ('í', '0000001'), ('ñ', '101100101'), ('ó', '101100100'), ('ú', '10110001100')]

Diccionario ordenado longitud código: [(' ', '110'), ('a', '011'), ('e', '010'), ('l', '0010'), ('n', '0001'), ('o', '1001'), ('r', '0011'), ('s', '1010'), ('c', '10000'), ('d', '11111'), ('i', '11110'), ('t', '10001'), ('u', '10111'), (',', '111001'), ('b', '111000'), ('m', '111011'), ('p', '111010'), ('q', '000001'), ('y', '000011'), ('g', '1011011'), ('h', '0000101'), ('í', '0000001'), ('.', '10110000'), ('f', '00000000'), ('j', '00001001'), ('v', '10110101'), ('z', '10110100'), ('á', '10110011'), (';', '000010001'), ('L', '000010000'), ('é', '000000011'), ('ñ', '101100101'), ('ó', '101100100'), ('B', '0000000100'), ('C', '1011000101'), ('E', '1011000100'), ('S', '0000000101'), ('N', '10110001111'), ('V', '10110001110'), ('x', '10110001101'), ('ú', '10110001100')]

Ratio de compresión: 1.8883351978033154 1856800 --> 983300

Ratio de compresión estimada con diccionario: 1.887076251377094 1856800 --> 983956

Diccionario a guardar: [[' ', 3], [',', 6], ['.', 8], [';', 9], ['B', 10], ['C', 10], ['E', 10], ['L', 9], ['N', 11], ['S', 10], ['V', 11], ['a', 3], ['b', 4], ['c', 4], ['d', 4], ['e', 4], ['f', 4], ['g', 4], ['h', 4], ['i', 4], ['j', 4], ['l', 4], ['m', 4], ['n', 4], ['o', 4], ['p', 4], ['q', 4], ['r', 4], ['s', 4], ['t', 4], ['u', 4], ['v', 4], ['x', 4], ['y', 4], ['z', 4], ['á', 4], ['é', 4], ['í', 4], ['ñ', 4], ['ó', 4], ['ú', 4]]
```


0.0.3 Codifico *La Regenta*

[71]:

Tiempo para crear tabla de frecuencias: 0.744138240814209
Entropía por símbolo: 4.421453902139422

Ratio de compresión máxima estimada sin diccionario: 1.8093595855718445

Tiempo para crear código Huffman: 0.001974344253540039

Tiempo para codificar: 0.17607498168945312

Diccionario ordenado alfabéticamente: [('\n', '110000'), (' ', '111'), ('!', '1011001100'), ('"', '10010000001100101'), ('(', '10110011111110'), (')', '10110011111101'), ('+', '1011001111111000111'), (',', '011101'), ('-', '110101010'), ('.', '100000'), ('0', '10110011111111000101'), ('1', '101100111111111011'), ('2', '101100111111110000'), ('3', '1001000000110011011'), ('4', '10010000001100110000'), ('5', '101100111111110001101'), ('6', '1011001111111100101'), ('7', '1001000000110011010'), ('8', '100100000011001111'), (':', '10010111101'), (';', '100101110'), ('?', '1001000111'), ('A', '011100000'), ('B', '101100111110'), ('C', '1001000101'), ('D', '1001000001'), ('E', '011100001'), ('F', '10110011010'), ('G', '100100000010'), ('H', '101100110110'), ('I', '100100000001'), ('J', '100100000000'), ('K', '1011001111111100'), ('L', '1001000100'), ('M', '1011001110'), ('N', '11010101111'), ('O', '10010110100'), ('P', '100100001'), ('Q', '10010111100'), ('R', '10110011110'), ('S', '1101010110'), ('T', '10010110101'), ('U', '1001000000111'), ('V', '1001011011'), ('W', '10110011111111010'), ('X', '100100000011000'), ('Y', '11010110100'), ('Z', '1011001111111101'), ('a', '010'), ('b', '100001'), ('c', '01111'), ('d', '10111'), ('e', '000'), ('f', '10110010'), ('g', '1011000'), ('h', '1001001'), ('i', '11001'), ('j', '01110001'), ('k', '1011001111111111'), ('l', '11011'), ('m', '110001'), ('n', '0010'), ('o', '1010'), ('p', '101101'), ('q', '1101000'), ('r', '0011'), ('s', '0110'), ('t', '10001'), ('u', '10011'), ('v', '1001010'), ('w', '101100111111110011'), ('x', '11010101110'), ('y', '0111001'), ('z', '110101110'), ('ą', '1001011111'), ('ż', '1001000000110011001'), ('ń', '11010110110'), ('ž', '100100000011001110'), ('ž', '11010110111'), ('&', '1001000110'), ('Á', '101100110111'), ('É', '10110011111100'), ('Í', '1011001111111100100'), ('Ů', '10010000001100110001'), ('Ů', '1001000000110010011'), ('à', '101100111111110001100'), ('á', '110101111'), ('ç', '101100111111110001001'), ('è', '100100000011001000'), ('é', '110101100'), ('í', '1101001'), ('ï', '101100111111110001000'), ('ñ', '100101100'), ('ó', '11010100'), ('ö', '1001000000110010010'), ('ú', '11010110101'), ('ü', '10010000001101')]

Diccionario ordenado longitud código: [(' ', '111'), ('a', '010'), ('e', '000'), ('n', '0010'), ('o', '1010'), ('r', '0011'), ('s', '0110'), ('c', '01111'), ('d', '10111'), ('i', '11001'), ('l', '11011'), ('t', '10001'), ('u', '10011'), ('\n', '110000'), (',', '011101'), ('.', '100000'), ('b', '100001'), ('m', '110001'), ('p', '101101'), ('g', '1011000'), ('h', '1001001'), ('q', '1101000'), ('v', '1001010'), ('y', '0111001'), ('í', '1101001'), ('f', '10110010'), ('j', '01110001'), ('ó', '11010100'), ('-', '110101010'), (';', '100101110'), ('A', '011100000'), ('E', '011100001'), ('P', '100100001'), ('z', '110101110'), ('á', '110101111'), ('é', '110101100'), ('ñ', '100101100'), ('!', '1011001100'), ('?', '1001000111'), ('C', '1001000101'), ('D', '1001000001'), ('L', '1001000100'), ('M', '1011001110'), ('S', '1101010110'), ('V',

```
'1001011011'), ('a', '1001011111'), ('&', '1001000110'), (':', '10010111101'),
('F', '10110011010'), ('N', '11010101111'), ('O', '10010110100'), ('Q',
'10010111100'), ('R', '10110011110'), ('T', '10010110101'), ('Y',
'11010110100'), ('x', '11010101110'), ('ñ', '11010110110'), ('ž',
'11010110111'), ('ú', '11010110101'), ('B', '101100111110'), ('G',
'100100000010'), ('H', '101100110110'), ('I', '100100000001'), ('J',
'100100000000'), ('Á', '101100110111'), ('U', '1001000000111'), ('(',
'10110011111110'), (')', '10110011111101'), ('É', '10110011111100'), ('ü',
'10010000001101'), ('X', '100100000011000'), ('Z', '1011001111111101'), ('k',
'1011001111111111'), ('"', '10010000001100101'), ('K', '10110011111111100'),
('1', '101100111111111011'), ('2', '101100111111110000'), ('8',
'100100000011001111'), ('W', '101100111111111010'), ('w', '101100111111110011'),
('ž', '100100000011001110'), ('è', '100100000011001000'), ('3',
'1001000000110011011'), ('6', '10110011111111100101'), ('7',
'1001000000110011010'), ('ž', '1001000000110011001'), ('í',
'10110011111111100100'), ('Ů', '1001000000110010011'), ('ö',
'1001000000110010010'), ('+', '101100111111111000111'), ('0',
'101100111111111000101'), ('4', '10010000001100110000'), ('ó',
'10010000001100110001'), ('5', '1011001111111110001101'), ('à',
'1011001111111110001100'), ('ç', '1011001111111110001001'), ('i',
'1011001111111110001000')]
```

Ratio de compresión: 1.793636939741042 14344224 --> 7997284 (bits)

Ratio de compresión estimada con diccionario: 1.7932889235383749 14344224 --> 7998836 (bits)

Tamaño Huffman : 999854 bytes

Tamaño Fichero .bz2: 518588 bytes

Tamaño Fichero .zip: 678394 bytes

Tamaño Fichero .gz: 710464 bytes

Tiempo Encode (Tabla frecuencias+Código Huffman+Codificación):
0.9228854179382324

Tiempo Decode: 0.8853530883789062

Hago distintas pruebas para ver como varía la ratio de compresión según el número de símbolos de la fuente tomados para calcular las frecuencias

```
[79]: with open ("../standard_test_text/la_regenta_utf8", "r") as myfile:
        mensaje=myfile.read()

for n_simbolos in range(1,5):
```

```

    ↪print('\n\n-----')
    print(' Codifico usando una fuente que toma '+str(n_simbolos)+ ' símbolos_
    ↪consecutivos')

    ↪
    ↪print('-----')

    t0=time.time()
    mensaje_codificado,
    ↪m2c,len_M=EncodeHuffman(mensaje,numero_de_simbolos=n_simbolos)
    t_encode=time.time()-t0
    ratio_compresion=8*len(mensaje)/len(mensaje_codificado)
    ratio_compresion_con_diccionario=8*len(mensaje)/
    ↪(len(mensaje_codificado)+len(m2c)*8*(n_simbolos+1))
    print('\n Ratio de compresión: ', ratio_compresion_
    ↪,8*len(mensaje),"-->",len(mensaje_codificado),'(bits)')
    print('\n Ratio de compresión estimada con diccionario: ',
    ↪ratio_compresion_con_diccionario,8*len(mensaje),"-->",len(mensaje_codificado)+len(m2c)*8*(n

    print('\n Tamaño Huffman      :
    ↪',int((len(mensaje_codificado)+len(m2c)*8*(n_simbolos+1))/8),'bytes')
    print('\n Tamaño Fichero .bz2: 518588 bytes')
    print('\n Tamaño Fichero .zip: 678394 bytes')
    print('\n Tamaño Fichero .gz: 710464 bytes')

    t0=time.time()
    mensaje_recuperado=DecodeHuffman(mensaje_codificado,m2c,len_M)
    t_decode=time.time()-t0
    print("\n Tiempo Encode (Tabla frecuencias+Código Huffman+Codificación):",
    ↪t_encode)
    print("\n Tiempo Decode:", t_decode)
    if (mensaje!=mensaje_recuperado):
        print('!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!')

```

Codifico usando una fuente que toma 1 símbolos consecutivos

Tiempo para crear tabla de frecuencias: 0.6799380779266357
Entropía por símbolo: 4.421453902139422
Ratio de compresión máxima estimada sin diccionario: 1.8093595855718445
Tiempo para crear código Huffman: 0.0017235279083251953
Tiempo para codificar: 0.1780836582183838

Ratio de compresión: 1.793636939741042 14344224 --> 7997284 (bits)

Ratio de compresión estimada con diccionario: 1.7932889235383749 14344224 -->
7998836 (bits)

Tamaño Huffman : 999854 bytes

Tamaño Fichero .bz2: 518588 bytes

Tamaño Fichero .zip: 678394 bytes

Tamaño Fichero .gz: 710464 bytes

Tiempo Encode (Tabla frecuencias+Código Huffman+Codificación):
0.8636305332183838

Tiempo Decode: 0.8202612400054932

Codifico usando una fuente que toma 2 símbolos consecutivos

Tiempo para crear tabla de frecuencias: 0.6237826347351074
Entropía por símbolo: 3.9270647087612174
Ratio de compresión máxima estimada sin diccionario: 2.037144939871281
Tiempo para crear código Huffman: 0.08007550239562988
Tiempo para codificar: 0.19551587104797363

Ratio de compresión: 2.030553808481733 14344224 --> 7064193 (bits)

Ratio de compresión estimada con diccionario: 2.0206339795063246 14344224 -->
7098873 (bits)

Tamaño Huffman : 887359 bytes

Tamaño Fichero .bz2: 518588 bytes

Tamaño Fichero .zip: 678394 bytes

Tamaño Fichero .gz: 710464 bytes

Tiempo Encode (Tabla frecuencias+Código Huffman+Codificación): 0.91117262840271

Tiempo Decode: 0.7272112369537354

Codifico usando una fuente que toma 3 símbolos consecutivos

Tiempo para crear tabla de frecuencias: 0.46556568145751953
Entropía por símbolo: 3.542521001298286
Ratio de compresión máxima estimada sin diccionario: 2.2582787786065652
Tiempo para crear código Huffman: 2.976529598236084
Tiempo para codificar: 0.1993095874786377

Ratio de compresión: 2.252255242605899 14344224 --> 6368827 (bits)

Ratio de compresión estimada con diccionario: 2.1488846393251433 14344224 --> 6675195 (bits)

Tamaño Huffman : 834399 bytes

Tamaño Fichero .bz2: 518588 bytes

Tamaño Fichero .zip: 678394 bytes

Tamaño Fichero .gz: 710464 bytes

Tiempo Encode (Tabla frecuencias+Código Huffman+Codificación):
4.068371057510376

Tiempo Decode: 0.6828961372375488

Codifico usando una fuente que toma 4 símbolos consecutivos

Tiempo para crear tabla de frecuencias: 0.4438338279724121
Entropía por símbolo: 3.2049578962185272
Ratio de compresión máxima estimada sin diccionario: 2.496132635451797
Tiempo para crear código Huffman: 35.41773271560669
Tiempo para codificar: 0.24241209030151367

Ratio de compresión: 2.4904792396170308 14344224 --> 5759624 (bits)

Ratio de compresión estimada con diccionario: 2.0153107272876456 14344224 --> 7117624 (bits)

Tamaño Huffman : 889703 bytes

Tamaño Fichero .bz2: 518588 bytes

Tamaño Fichero .zip: 678394 bytes

Tamaño Fichero .gz: 710464 bytes

Tiempo Encode (Tabla frecuencias+Código Huffman+Codificación):

41.28862428665161

Tiempo Decode: 0.7026417255401611

[]: