Huffman Encoding

Rodrigo Castillo junto a Nicolás Otero

5 de octubre de 2020

1. Huffman Encoding para $TEORIADEGRAFOS$

1.1. creación de la tabla

la palabra $TEORIADEGRAFOS$ se puede expresar como un diccionario de la forma $dic = \{[T,1][E,2],[R,2],[I,1],[A,2],[D,1],[G,1],[F,1],[O,1],[S,1]\}$
teniendo el diccionario, podemos ordenarlo, debido a que hay tantas letras con el mismo peso, no importa mucho el orden, sin embargo, el orden quedaría así:

1. $[A,2]$
2. $[E,2]$
3. $[R,2]$
4. $[T,1]$
5. $[I,1]$
6. $[D,1]$
7. $[G,1]$
8. $[F,1]$
9. $[O,1]$
10. $[S,1]$

Para codificar el string $TEORIADEGRAFOS$ vamos a suponer que el string está originalmente cifrado en
$ASCII(TEORIADEGRAFOS) =$. 45 54 52 $4f$ 41 49 45 44 52 47 46 41 53 $4f$.
, ahora recordaremos esto para compararlo con el ejemplo codificado mediante el algoritmo de huffman.
mediante el algoritmo de huffman, tomaremos el root como el elemento 1 y construiremos el arbol con respecto a este, por lo tanto .

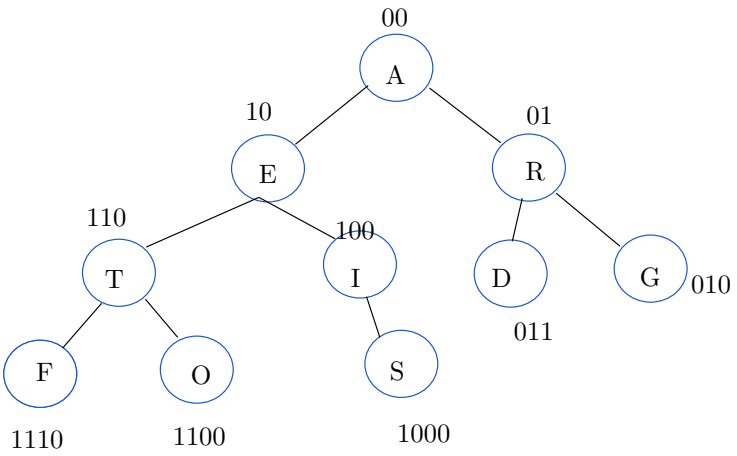

Figura 1: Arbol de Huffman para string $TEORIADEGRAFOS$

por lo tanto ahora el string es 110 10 1100 01 100 00 011 10 010 01 00 1110 1100 1000
note que :
ASCCI(TEORIADEGRAFOS) = 45 54 52 4f 41 49 45 44 52 47 46 41 53 4f
BIN(ASCCI(TEORIADEGRAFOS)) = 1000101 1010100 1010010 1001111 1000001
1001001 1000101 1000100 1010010 1000111 1000110 1000001 1010011 1001111 :
y esto puede verse que es mucho menos pesado que
110 10 1100 01 100 00 011 10 010 01 00 1110 1100 1000
sin embargo el string resultante de la codificacion huffman puede no tener sentido como binario , sin embargo, es el siguiente:
Codificación de Huffman para el string $TEORIADEGRAFOS$ = 11010110 00110000 01110010 01001110 11001000.