Algoritmo de Huffman

Caso Práctico

Dado un archivo que contiene la siguiente frase:

TOMAMOS COMO EJEMPLO ESTA FRASE

Se procederá a implementar el algoritmo de Huffman, generando la tabla de frecuencias y el árbol binario de búsqueda. Luego de ello se compactará dicho archivo.

La implementación del mismo consta de cuatro etapas,

- Generación de la tabla de Frecuencias
- Generación del Arbol Binario
- Compresión o compactación de datos.
- Descompresión o descompactación de datos.

las dos primeras etapas hacen a la construcción de las estructuras de memoria, y las dos últimas al tratamiento de los datos.

Generación de la tabla de frecuencias

Se realiza un proceso de conteo de los caracteres del archivo y se van cargando en una estructura de memoria que contiene a la tabla de frecuencias (puede ser una lista linkeada en la cual cada nodo cuenta con los campos necesarios de la tabla).

Para facilitar la explicación, procedimos a ordenar los elementos de la tabla de frecuencia de mayor a menor.

Status	*Char*	*Freq*	*Code*	*Dir_tree*
	Blanco	5		
	O	5		
	M	4		
	E	4		
	A	3		
	S	3		
	T	2		
	C	1		
	J	1		
	P	1		
	L	1		
	F	1		
	R	1		

La cantidad total de caracteres es 32 x 8 bits = 256 bits

Generación del Arbol Binario

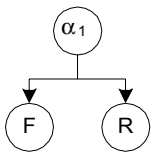
Para armar el árbol se respetó el concepto de tomar de la tabla de frecuencias los elementos más chicos inicialmente, pero se definieron además las siguientes reglas:

1. Se toman primero los menores caracteres tomados de a pares, comenzando por los dos últimos.
2. El par de valores se toman de la sig. manera:
   - Si ambos tienen igual frecuencia, se toma como hijo izquierdo el primero y como hijo derecho al último.
   - Si uno tiene mayor frecuencia, se toma como hjjo izquierdo el de menor frecuencia y como hijo derecho el de mayor frecuencia.
3. Siempre se toman primero los caracteres originales del archivo de menor frecuencia, sin no hubiesen más o hubiesen de frecuencia mayor, se procederán a tomar los caracteres "α_x".

Paso 1

Tomo los dos primeros caracteres de menor frecuencia F y R. Armo un sub-árbol con raíz "α_1" en memoria con ambos nodos F como hijo izq. y R como hijo derecho. Actualizo la tabla de frecuencias con el nuevo nodo "α_1" y con su frecuencia como la suma de frecuencias de sus hijos, las direcciones de los tres nodos y el status de F y R.

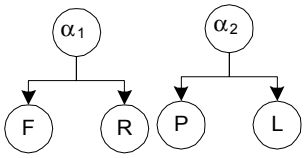
Status	*Char*	*Freq*	*Code*	*Dir_tree*
	Blanco	5		
	O	5		
	M	4		
	E	4		
	A	3		
	S	3		
	T	2		
	C	1		
	J	1		
	P	1		
	L	1		
x	F	1		**dir (F)**
x	R	1		**dir (R)**
	α_1	**2**		**dir (α_1)**



Paso 2

Tomo los próximos dos caracteres de menor frecuencia, P y L. Armo un sub-árbol con raíz "α_2" en memoria con ambos nodos P como hijo izq. y L como hijo derecho. Actualizo la tabla de frecuencias con el nuevo nodo "α_2" y con su frecuencia como la suma de frecuencias de sus hijos, las direcciones de los tres nodos y el status de P y L.

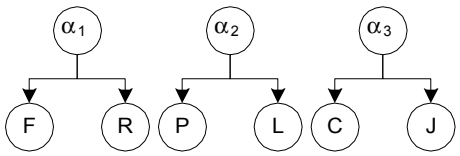
Status	***Char***	***Freq***	***Code***	***Dir_tree***
	Blanco	5		
	O	5		
	M	4		
	E	4		
	A	3		
	S	3		
	T	2		
	C	1		
	J	1		
x	P	1		**dir (P)**
x	L	1		**dir (L)**
x	F	1		dir (F)
x	R	1		dir (R)
	α_1	2		dir (α_1)
	α_2	**2**		**dir (α_2)**



Paso 3

Tomo los próximos dos caracteres de menor frecuencia, C y J. Armo un sub-árbol con raíz "α_3" en memoria con ambos nodos C como hijo izq. y J como hijo derecho. Actualizo la tabla de frecuencias con el nuevo nodo "α_3" y con su frecuencia como la suma de frecuencias de sus hijos, las direcciones de los tres nodos y el status de C y J.

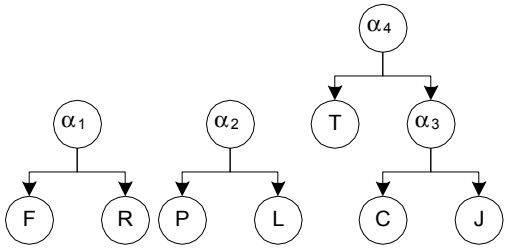
Status	***Char***	***Freq***	***Code***	***Dir_tree***
	Blanco	5		
	O	5		
	M	4		
	E	4		
	A	3		
	S	3		
	T	2		
x	C	1		**dir (C)**
x	J	1		**dir (J)**
x	P	1		dir (P)
x	L	1		dir (L)
x	F	1		dir (F)
x	R	1		dir (R)
	α_1	2		dir (α_1)
	α_2	2		dir (α_2)
	α_3	**2**		**dir (α_3)**



Paso 4

Tomo los próximos dos caracteres de menor frecuencia, T y α_3. Armo un sub-árbol con raíz "α_4" en memoria con ambos nodos T como hijo izq. y α_3 como hijo derecho. Actualizo la tabla de frecuencias con el nuevo nodo "α_4" y con su frecuencia como la suma de frecuencias de sus hijos, las direcciones del nodo raíz y T, y el status de T y α_3.

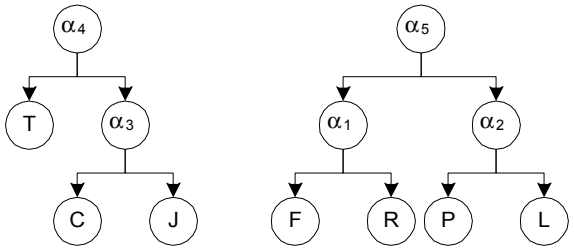
Status	***Char***	***Freq***	***Code***	***Dir_tree***
	Blanco	5		
	O	5		
	M	4		
	E	4		
	A	3		
	S	3		
x	T	2		**dir (T)**
x	C	1		dir (C)
x	J	1		dir (J)
x	P	1		dir (P)
x	L	1		dir (L)
x	F	1		dir (F)
x	R	1		dir (R)
	α_1	2		dir (α_1)
	α_2	2		dir (α_2)
x	α_3	2		dir (α_3)
	α_4	**4**		**dir (α_4)**



Paso 5

Tomo los próximos dos caracteres de menor frecuencia, α_1 y α_2. Armo un sub-árbol con raíz "α_5" en memoria con ambos nodos α_1 como hijo izq. y α_2 como hijo derecho. Actualizo la tabla de frecuencias con el nuevo nodo "α_5" y con su frecuencia como la suma de frecuencias de sus hijos, las direcciones del nodo raíz y el status de α_1 y α_2.

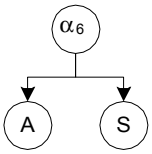
Status	***Char***	***Freq***	***Code***	***Dir_tree***
	Blanco	5		
	O	5		
	M	4		
	E	4		
	A	3		
	S	3		
x	T	2		dir (T)
x	C	1		dir (C)
x	J	1		dir (J)
x	P	1		dir (P)
x	L	1		dir (L)
x	F	1		dir (F)
x	R	1		dir (R)
x	α_1	2		dir (α_1)
x	α_2	2		dir (α_2)
x	α_3	2		dir (α_3)
	α_4	4		dir (α_4)
	α_5	**4**		**dir (α_5)**

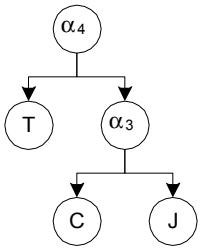


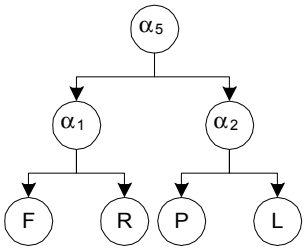
Paso 6

Tomo los próximos dos caracteres de menor frecuencia, A y S. Armo un sub-árbol con raíz "α_6" en memoria con ambos nodos A como hijo izq. y S como hijo derecho. Actualizo la tabla de frecuencias con el nuevo nodo "α_6" y con su frecuencia como la suma de frecuencias de sus hijos, las direcciones de los tres nodos y el status de A y S.

Status	***Char***	***Freq***	***Code***	***Dir_tree***
	Blanco	5		
	O	5		
	M	4		
	E	4		
x	A	3		**dir (A)**
x	S	3		**dir (S)**
x	T	2		dir (T)
x	C	1		dir (C)
x	J	1		dir (J)
x	P	1		dir (P)
x	L	1		dir (L)
x	F	1		dir (F)
x	R	1		dir (R)
x	α_1	2		dir (α_1)
x	α_2	2		dir (α_2)
x	α_3	2		dir (α_3)
	α_4	4		dir (α_4)
	α_5	4		dir (α_5)
	α_6	**6**		**dir (α_6)**



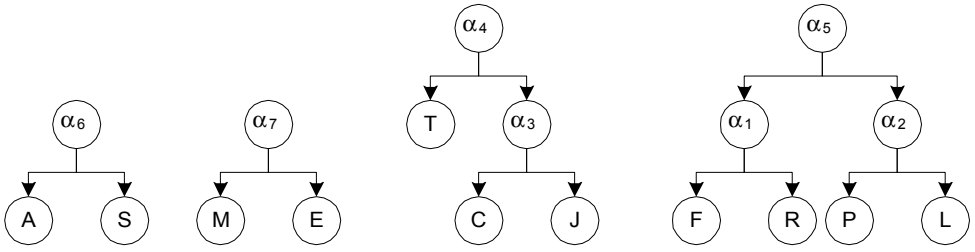




Paso 7

Tomo los próximos dos caracteres de menor frecuencia, M y E. Armo un sub-árbol con raíz "α_7" en memoria con ambos nodos M como hijo izq. y E como hijo derecho. Actualizo la tabla de frecuencias con el nuevo nodo "α_7" y con su frecuencia como la suma de frecuencias de sus hijos, las direcciones de los tres nodos y el status de M y E.

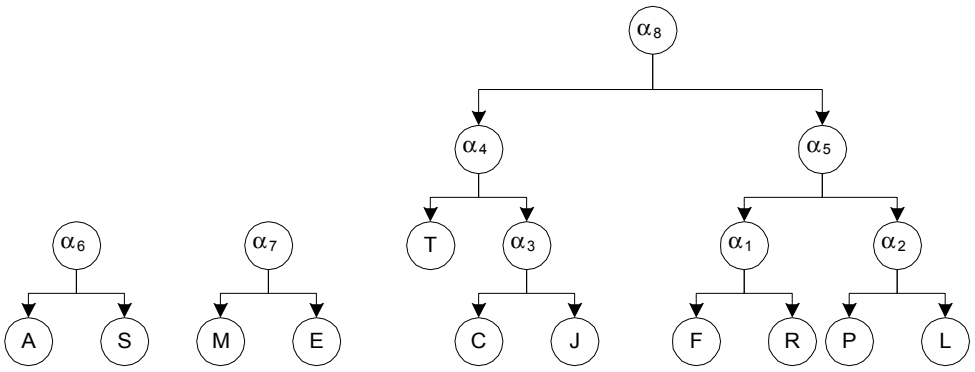
Status	***Char***	***Freq***	***Code***	***Dir_tree***
	Blanco	5		
	O	5		
x	M	4		**dir (M)**
x	E	4		**dir (E)**
x	A	3		dir (A)
x	S	3		dir (S)
x	T	2		dir (T)
x	C	1		dir (C)
x	J	1		dir (J)
x	P	1		dir (P)
x	L	1		dir (L)
x	F	1		dir (F)
x	R	1		dir (R)
x	α_1	2		dir (α_1)
x	α_2	2		dir (α_2)
x	α_3	2		dir (α_3)
	α_4	4		dir (α_4)
	α_5	4		dir (α_5)
	α_6	6		dir (α_6)
	α_7	**8**		**dir (α_7)**



Paso 8

Tomo los próximos dos caracteres de menor frecuencia, α_4 y α_5. Armo un sub-árbol con raíz "α_8" en memoria con ambos nodos α_4 como hijo izq. y α_5 como hijo derecho. Actualizo la tabla de frecuencias con el nuevo nodo "α_8" y con su frecuencia como la suma de frecuencias de sus hijos, las direcciones del nodo raiz y el status de α_4 y α_5.

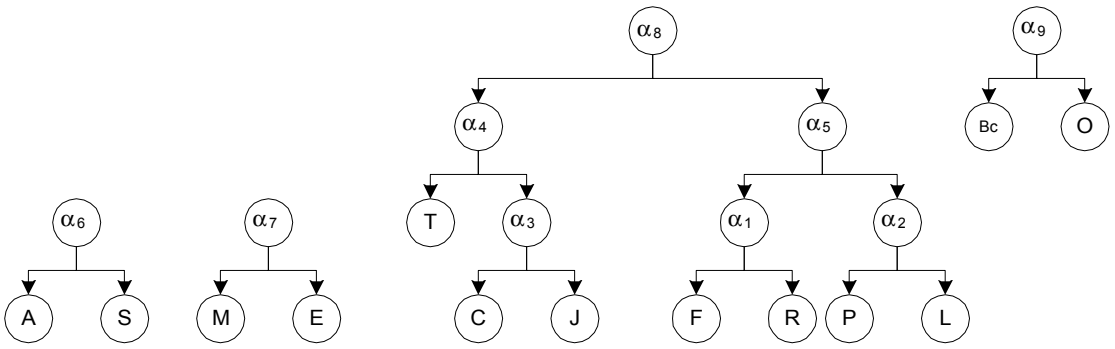
Status	***Char***	***Freq***	***Code***	***Dir_tree***
	Blanco	5		
	O	5		
x	M	4		dir (M)
x	E	4		dir (E)
x	A	3		dir (A)
x	S	3		dir (S)
x	T	2		dir (T)
x	C	1		dir (C)
x	J	1		dir (J)
x	P	1		dir (P)
x	L	1		dir (L)
x	F	1		dir (F)
x	R	1		dir (R)
x	α_1	2		dir (α_1)
x	α_2	2		dir (α_2)
x	α_3	2		dir (α_3)
x	α_4	4		dir (α_4)
x	α_5	4		dir (α_5)
	α_6	6		dir (α_6)
	α_7	8		dir (α_7)
	α_8	**8**		**dir (α_8)**



Paso 9

Tomo los próximos dos caracteres de menor frecuencia, Blanco y O. Armo un sub-árbol con raíz "α_9" en memoria con ambos nodos Blanco como hijo izq. y O como hijo derecho. Actualizo la tabla de frecuencias con el nuevo nodo "α_9" y con su frecuencia como la suma de frecuencias de sus hijos, las direcciones de los tres nodos y el status de Blanco y O.

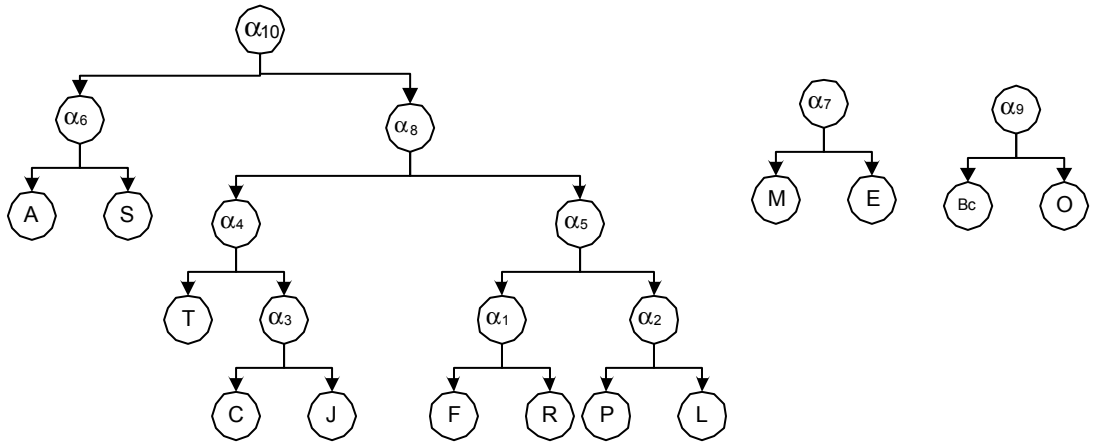
Status	***Char***	***Freq***	***Code***	***Dir_tree***
x	Blanco	5		**dir (Bco)**
x	O	5		**dir (O)**
x	M	4		dir (M)
x	E	4		dir (E)
x	A	3		dir (A)
x	S	3		dir (S)
x	T	2		dir (T)
x	C	1		dir (C)
x	J	1		dir (J)
x	P	1		dir (P)
x	L	1		dir (L)
x	F	1		dir (F)
x	R	1		dir (R)
x	α_1	2		dir (α_1)
x	α_2	2		dir (α_2)
x	α_3	2		dir (α_3)
x	α_4	4		dir (α_4)
x	α_5	4		dir (α_5)
	α_6	6		dir (α_6)
	α_7	8		dir (α_7)
	α_8	8		dir (α_8)
	α_9	**10**		**dir (α_9)**



Paso 10

Tomo los próximos dos caracteres de menor frecuencia, α_6 y α_7. Armo un sub-árbol con raíz "α_{10}" en memoria con ambos nodos α_6 como hijo izq. y α_7 como hijo derecho. Actualizo la tabla de frecuencias con el nuevo nodo "α_{10}" y con su frecuencia como la suma de frecuencias de sus hijos, las direcciones del nodo raíz y el status de α_6 y α_7.

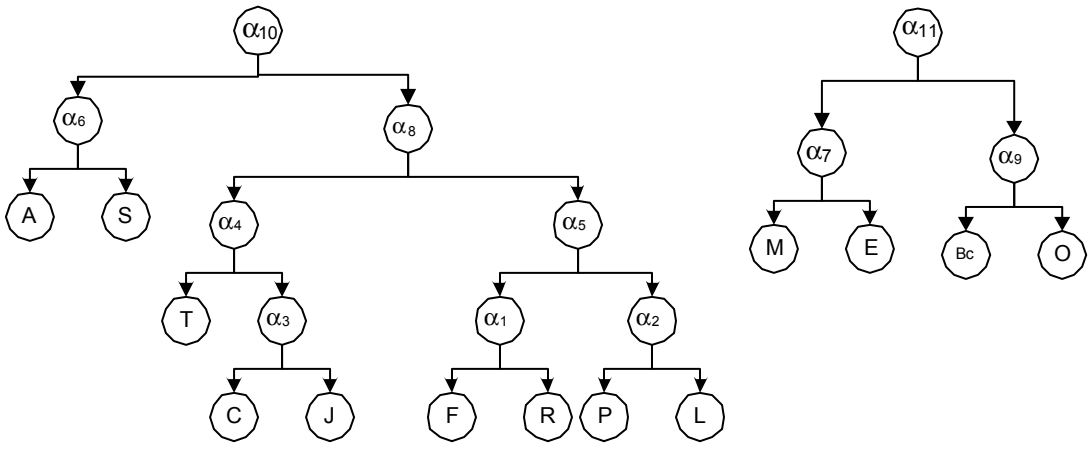
Status	***Char***	***Freq***	***Code***	***Dir_tree***
x	Blanco	5		dir (Bco)
x	O	5		dir (O)
x	M	4		dir (M)
x	E	4		dir (E)
x	A	3		dir (A)
x	S	3		dir (S)
x	T	2		dir (T)
x	C	1		dir (C)
x	J	1		dir (J)
x	P	1		dir (P)
x	L	1		dir (L)
x	F	1		dir (F)
x	R	1		dir (R)
x	α_1	2		dir (α_1)
x	α_2	2		dir (α_2)
x	α_3	2		dir (α_3)
x	α_4	4		dir (α_4)
x	α_5	4		dir (α_5)
x	α_6	6		dir (α_6)
	α_7	8		dir (α_7)
x	α_8	8		dir (α_8)
	α_9	10		dir (α_9)
	α_{10}	**14**		**dir (α_{10})**



Paso 11

Tomo los próximos dos caracteres de menor frecuencia, α_8 y α_9. Armo un sub-árbol con raíz "α_{11}" en memoria con ambos nodos α_8 como hijo izq. y α_9 como hijo derecho. Actualizo la tabla de frecuencias con el nuevo nodo "α_{11}" y con su frecuencia como la suma de frecuencias de sus hijos, las direcciones del nodo raíz y el status de α_8 y α_9.

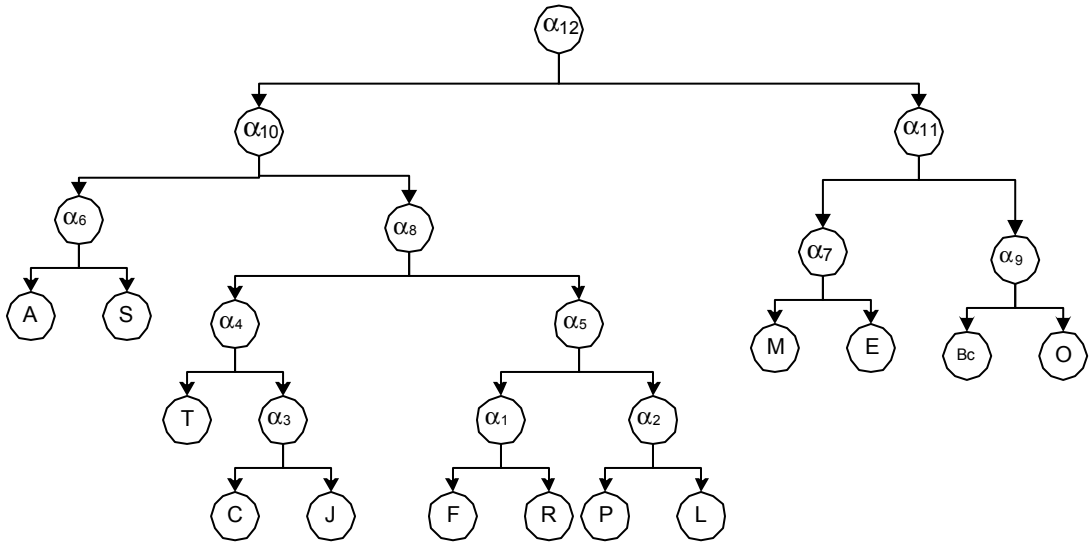
Status	***Char***	***Freq***	***Code***	***Dir_tree***
x	Blanco	5		dir (Bco)
x	O	5		dir (O)
x	M	4		dir (M)
x	E	4		dir (E)
x	A	3		dir (A)
x	S	3		dir (S)
x	T	2		dir (T)
x	C	1		dir (C)
x	J	1		dir (J)
x	P	1		dir (P)
x	L	1		dir (L)
x	F	1		dir (F)
x	R	1		dir (R)
x	α_1	2		dir (α_1)
x	α_2	2		dir (α_2)
x	α_3	2		dir (α_3)
x	α_4	4		dir (α_4)
x	α_5	4		dir (α_5)
x	α_6	6		dir (α_6)
x	α_7	8		dir (α_7)
x	α_8	8		dir (α_8)
x	α_9	10		dir (α_9)
	α_{10}	14		dir (α_{10})
	α_{11}	**18**		**dir (α_{11})**



Paso 12

Tomo los próximos dos caracteres de menor frecuencia, α_{10} y α_{11}. Armo un sub-árbol con raíz "α_{12}" en memoria con ambos nodos α_{10} como hijo izq. y α_{11} como hijo derecho. Actualizo la tabla de frecuencias con el nuevo nodo "α_{12}" y con su frecuencia como la suma de frecuencias de sus hijos, las direcciones del nodo raíz y el status de α_{10} y α_{11}.

Status	***Char***	***Freq***	***Code***	***Dir_tree***
x	Blanco	5		dir (Bco)
x	O	5		dir (O)
x	M	4		dir (M)
x	E	4		dir (E)
x	A	3		dir (A)
x	S	3		dir (S)
x	T	2		dir (T)
x	C	1		dir (C)
x	J	1		dir (J)
x	P	1		dir (P)
x	L	1		dir (L)
x	F	1		dir (F)
x	R	1		dir (R)
x	α_1	2		dir (α_1)
x	α_2	2		dir (α_2)
x	α_3	2		dir (α_3)
x	α_4	4		dir (α_4)
x	α_5	4		dir (α_5)
x	α_6	6		dir (α_6)
x	α_7	8		dir (α_7)
x	α_8	8		dir (α_8)
x	α_9	10		dir (α_9)
x	α_{10}	14		dir (α_{10})
x	α_{11}	18		dir (α_{11})
	α_{12}	**32**		**dir (α_{12})**



Paso 13

Al querer tomar los próximos dos caracteres de menor frecuencia, se observa que queda un único carácter con status null y que su frecuencia es la suma de todas las frecuencias de los caracteres originales. Por lo tanto se da por finalizado la generación del árbol.

Compresión o Compactación de Datos.

Este proceso comprende la lectura del archivo original, la búsqueda en la tabla de frecuencias de dicho carácter leído, y el acceso con la dirección del árbol a la hoja correspondiente a dicho carácter. A partir de ello se va ascendiendo por el árbol con el puntero al padre (utilizando para almacenar un 0 si asciendo desde un hijo izq. ó un 1 si asciendo del hijo derecho) hasta llegar al nodo raíz (padre en null). En ese momento se desapilará la pila en una estructura de salida, y se ingresará sólo para la primera lectura de cada carácter el código comprimido en la tabla de frecuencias, para optimizar el algoritmo.

Al finalizar este proceso la tabla de frecuencias quedará con la siguiente información:

Status	***Char***	***Freq***	***Code***	***Dir_tree***
x	Blanco	5	110	dir (Bco)
x	O	5	111	dir (O)
x	M	4	100	dir (M)
x	E	4	101	dir (E)
x	A	3	000	dir (A)
x	S	3	001	dir (S)
x	T	2	0100	dir (T)
x	C	1	01010	dir (C)
x	J	1	01011	dir (J)
x	P	1	01110	dir (P)
x	L	1	01111	dir (L)
x	F	1	01100	dir (F)
x	R	1	01101	dir (R)
x	α_1	2		dir (α_1)
x	α_2	2		dir (α_2)
x	α_3	2		dir (α_3)
x	α_4	4		dir (α_4)
x	α_5	4		dir (α_5)
x	α_6	6		dir (α_6)
x	α_7	8		dir (α_7)
x	α_8	8		dir (α_8)
x	α_9	10		dir (α_9)
x	α_{10}	14		dir (α_{10})
x	α_{11}	18		dir (α_{11})
	α_{12}	**32**		**dir (α_{12})**

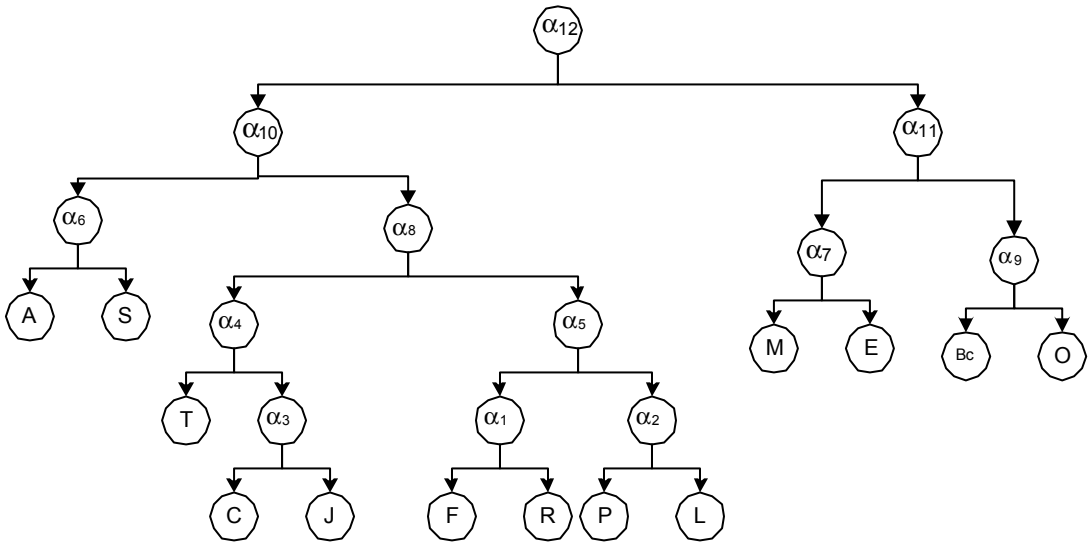
El archivo comprimido sería:

0100111100000100111001110010101111001111101010101110110001110011111111101010010
1000001100110001101000001101

En total serían 107 bits, que en comparación con los 256 bits originales observamos una compactación de aprox. 42%.

Descompresión o Descompactación de datos

Este proceso consiste en ir leyendo los bits del archivo compactado y accediendo desde la raíz del árbol ir descendiendo por el mismo, de la siguiente forma: si leo un bit 0 se deberá descender por el hijo izquierdo, si en cambio, leo un bit 1 se deberá descender por el hijo derecho. Este descenso será hasta llegar a una hoja (hijos en null), en dicho caso se informará su Id. Y se volverá a la raíz para seguir parseando los próximos bits.



0100 111 100 000 100 111 001 110 01010 111 100 111 110 101 01011 101 100 01110 01111 111
T O M A M O S __ C O M O __ E J E M P L O
110 101 001 0100 000 110 01100 01101 000 001 101
__ E S T A __ F R A S E