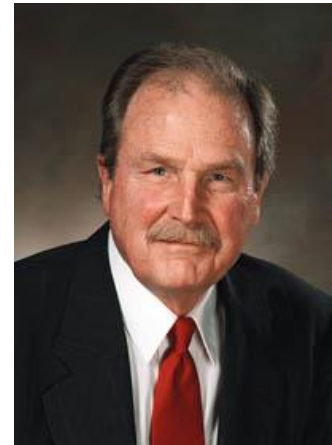
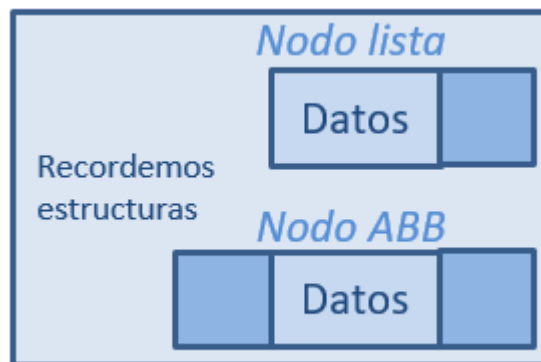


Ejemplo de codificación huffman

Codificación Huffman

Ejemplo de codificación (*paso a paso*):

Usamos una estructura mixta, una lista de árboles.



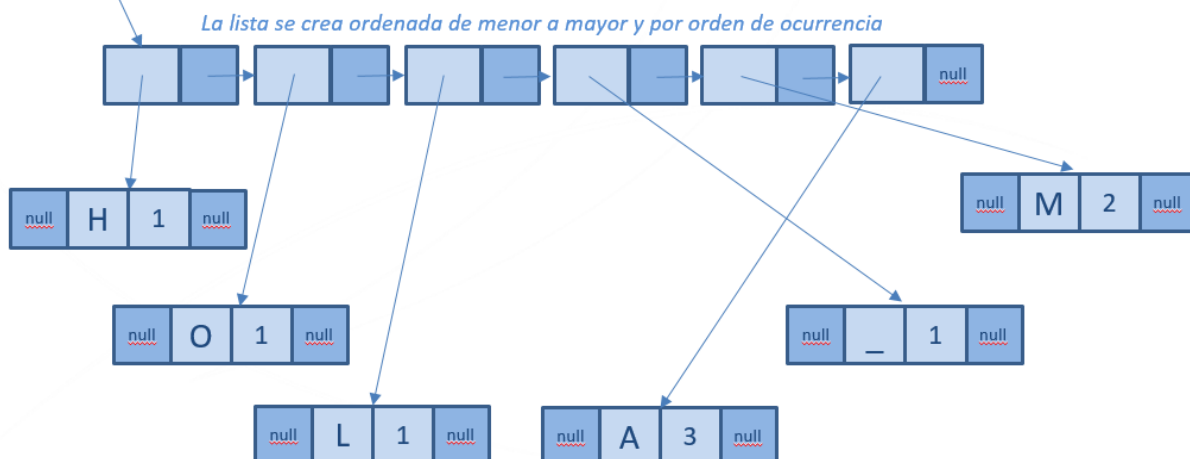
Mensaje: **HOLA MAMA**

Es
Espacio

9 caracteres a codificar

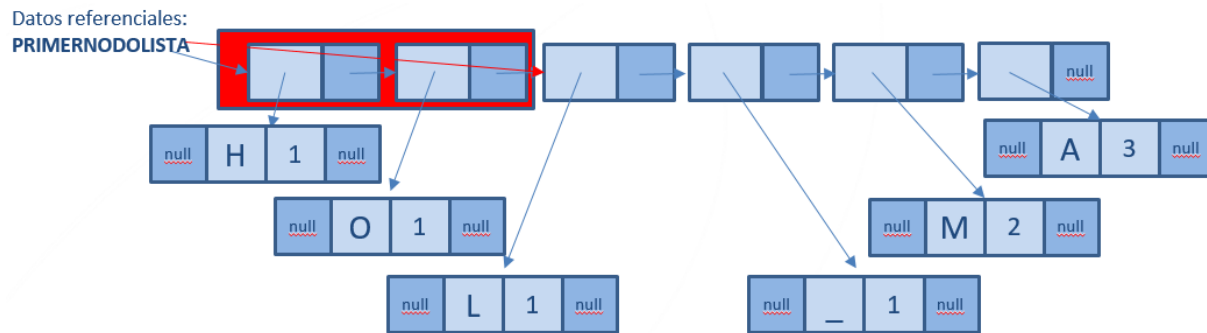
1) Armamos una lista de nodos de árbol binario (*ordenada por ocurrencias de menor a mayor*)

Datos referenciales:
PRIMER NODO LISTA

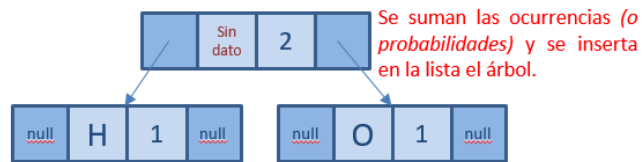


2) Iniciamos el proceso de armado de sub arboles de 2 en 2 (*primera iteración*)

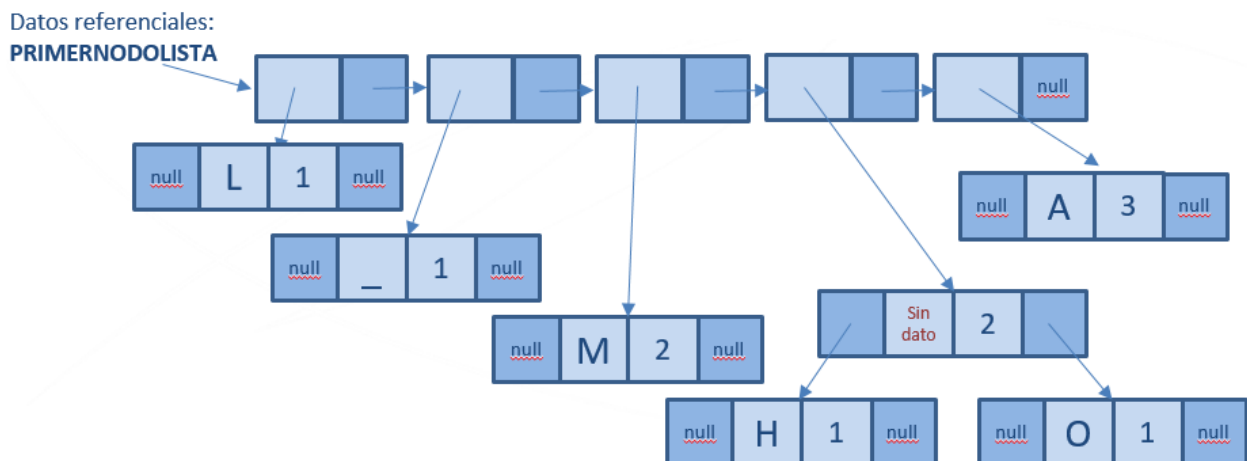
Eliminamos “momentáneamente” los 2 primeros nodos de la lista.



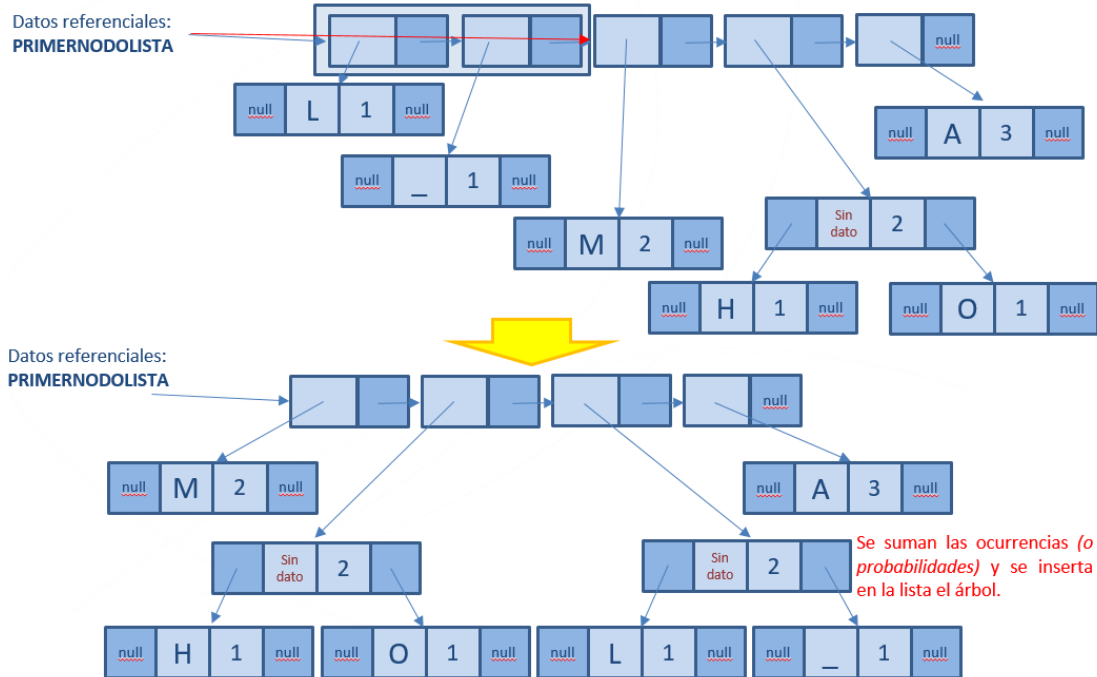
Decimos momentáneamente, porque le creamos un nodo padre a ambos sumándole las ocurrencias de los hijos



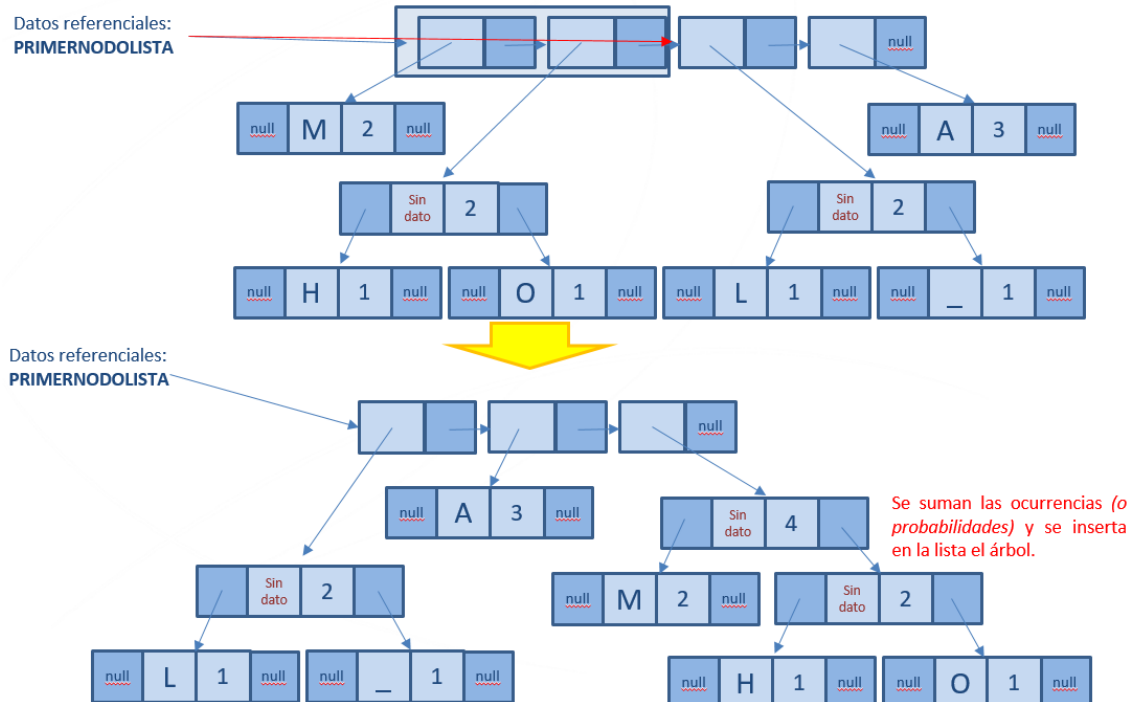
Y luego a ese nodo padre lo volvemos a insertar en la lista



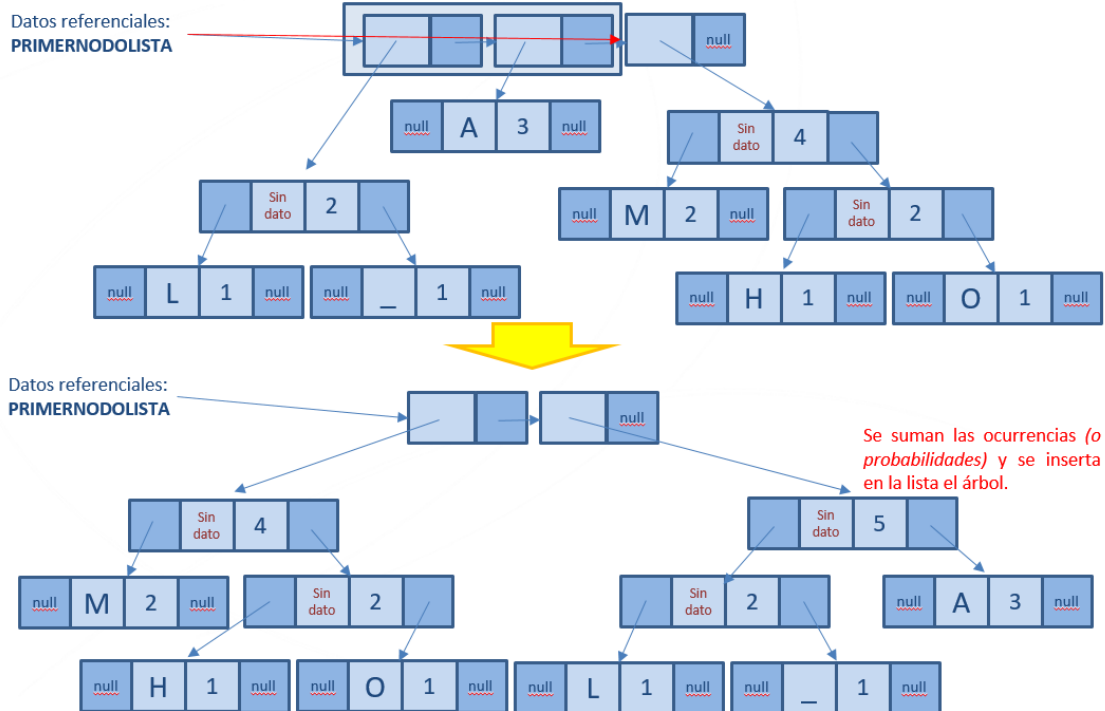
2da iteración del proceso de armado de sub arboles de 2 en 2
Volvemos a eliminar “momentáneamente” los 2 primeros nodos de la lista.



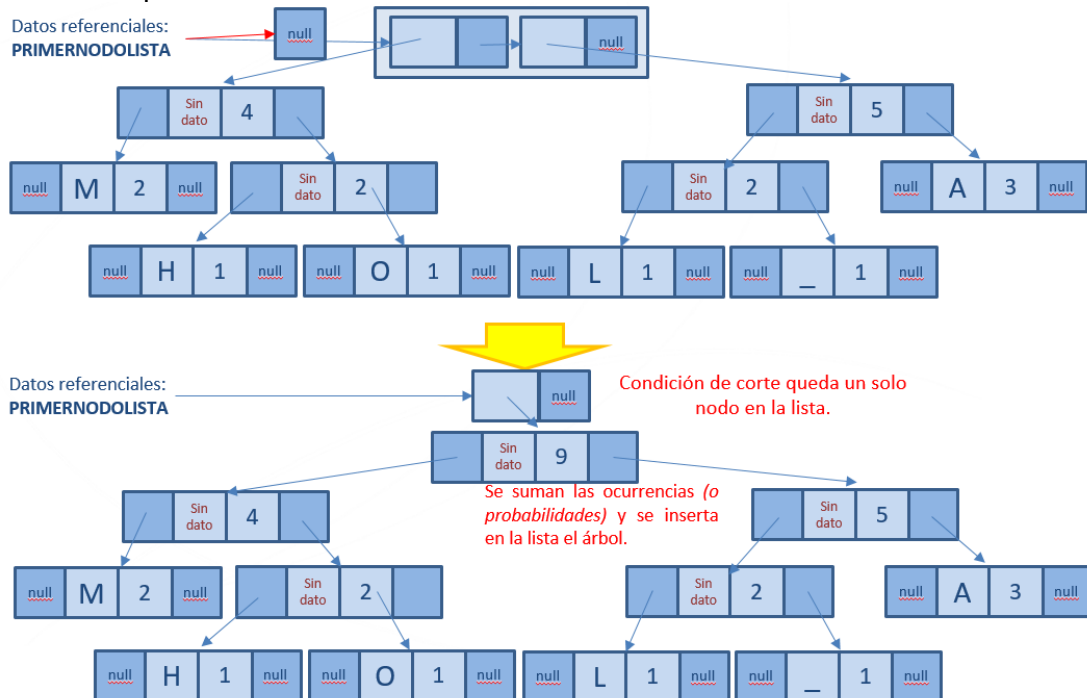
3er iteración del proceso de armado de sub arboles de 2 en 2



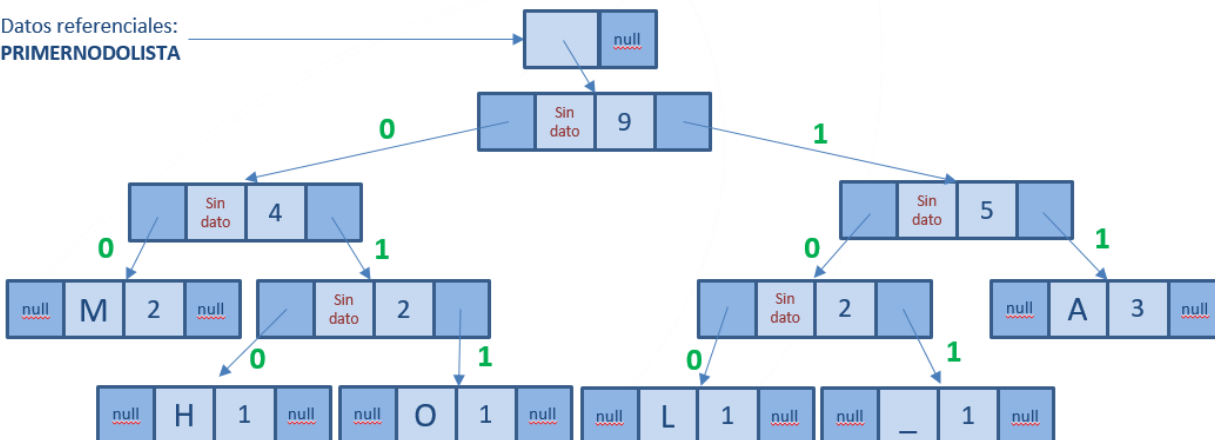
4ta iteración del proceso de armado de sub arboles de 2 en 2



5ta iteración del proceso de armado de sub arboles de 2 en 2



Datos referenciales:
PRIMERNODOLISTA



H	010	72	01001000
O	011	79	01001111
L	100	76	01001100
—	101	32	00100000
M	00	77	01001101
A	11	65	01000001

```
01001000 01001111 01001100 01000001
00100000 01001101 01000001 01001101 01000001
```

Mensaje binario compactado
0100111001110100110011

Mensaje binario compactado Teórico

0100111001110100110011

Mensaje binario compactado en forma práctica

01001110	01110100	11001100
----------	----------	----------

BITS de relleno para completar el byte

5) Generamos el archivo original (*descomprimido*) a partir del archivo codificado (*comprimido*) usando la tabla. **DESCOMPRESIÓN**

NOTAS: * No se puede descomprimir el archivo si no poseemos la tabla y/o el árbol.

* Precisamos saber cuantos símbolos había en el archivo original (cuestiones de implementación).

Se concatenan bits en una variable auxiliar hasta que sea una de las entradas de la tabla

Mensaje binario compactado

010011100111010011001100

Mensaje ascii original:

Mensaje binario original

010010000100111101001100010000010010000001001101010000010100110101000001

Si no hubiésemos contado los símbolos del archivo original, confundiríamos los bits de relleno con información a decodificar.

Para implementar descompresión por **Huffman**, deberemos haber guardado previamente, tanto la información de la tabla con cantidad de ocurrencias y/o árbol, como la cantidad de símbolos que poseía el archivo sin comprimir. De lo contrario nos sería imposible descomprimir el mensaje.