1. Huffman Encoding Programming
   1. Example:

n2: number of nodes of degree 2.

n1: number of nodes of degree 1, n1=0 when it in a Huffman tree.

n0: number of leaves, also the number of symbols.



Figure : A Huffman Tree Consisting of Five Symbols

Total nodes=n1+n2+n0=n2\*2+n1+1

->n0=n2+1

Nt=2(n0-1)+n0=3n0-2, nt is the entry number of Huffman table

We use 0,1,2,3,4 to represent a,b,c,d,e.

Huffman Table:

|  |  |  |
| --- | --- | --- |
| Index | Table Content | Comments |
| 12 | 0 | ‘a’ is saved here, left child of node 0 |
| 11 | 10 | The right child of node 0, |
| 10 | 9 | Left child of node 1, |
| 9 | 4 | ‘e’ is saved here, right child of node 1, |
| 8 | 1 | ‘b’ is saved here, left child of node 2, |
| 7 | 6 | Right child of node 2, |
| 6 | 2 | ‘c’ is saved here, left child of node 3 |
| 5 | 3 | ‘d’ is saved here, right child of node 3, |
| 4 | e | Symbol at leave |
| 3 | d | Symbol at leave |
| 2 | c | Symbol at leave |
| 1 | b | Symbol at leave |
| 0 | a | Symbol at leave |

Code of this Huffman table.

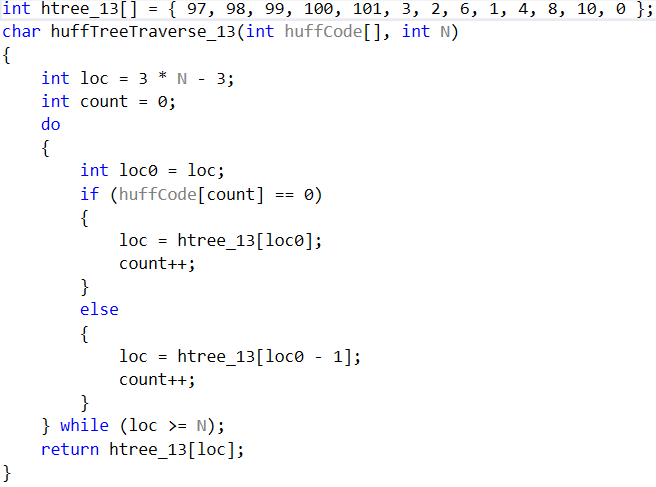


Figure : c++ code about huffman encoding

Huffman table can be simplified, we use a 2 bytes word. Upper 8 bits are used to save left child and lower 8bits are used to save right child. The entries can be reduced to 4 (which is the number of nodes).

|  |  |  |  |
| --- | --- | --- | --- |
| Index | Left child | Right child | Content |
| 3(8) | 0 (a) | 7 | Left and right child of node 0 (root) |
| 2(7) | 6 | 4 (e) | Left and right child of node 1 |
| 1(6) | 1 (b) | 5 | Left and right child of node 2 |
| 0(5) | 2 (c) | 3(d) | Left and right child of node 3 |

3(8): 8 is the node number.

Childs of node 0 can be saved in a short variable. E.g. 0x0007

