CST Ib, CST II(G) and Diploma

Data Structures and Algorithms

by

Martin Richards

mr@cl.cam.ac.uk

http://www.cl.cam.ac.uk/users/mr/

University Computer Laboratory
New Museum Site
Pembroke Street
Cambridge, CB2 3QG

Ternary Trees

Adding the word:

`MIT, SAD, MAN, APT, MUD, ADD, MAG, MINE, MIKE, MINT, AT, MATE, MINES`

produces the following tree:

```
                                            |
       -------------------------------------M--------
      |                                     |         |
     -A-              ----------------------I----    -S-
      |              |                      |    |    |
  ----P----         -A-              -------T-  -U-  -A-
 |    |    |         |              |       |    |    |
-D-  -T-  -T-    ----N----      ----N-     -*-  -D-  -D-
 |    |    |    |    |    |    |    |            |    |
-D-  -*-  -*-  -G-  -*-  -T-  -K-  -E-          -*-  -*-
 |              |         |    |    |
-*-            -*-       -E-  -E-  -*---
                          |    |        |
                         -*-  -*-      -S-
                                        |
                                       -*-
```

Huffman

```
============leaves=============  ==============nodes===========
P  F  B  L  E  T  J  Q  K  A  G (symbols)
1  1  1  1  1  1  1  1  2  4  5 (frequencies)
#  #  #  #  #  #  #  #  #  #  #
|  |  |  |  |  |  |  |  |  |  |               node types
 --+---------------------------(2)LL
4  4  |  |  |  |  |  |  |  |  | 3#            LL = leaf+leaf
       --+------------------------(2)LL       LN = leaf+node
      4  4  |  |  |  |  |  |  |  | 3#         NN = node+node
             --+---------------------(2)LL
            4  4  |  |  |  |  |  |  | 3#
                   --+------------------(2)LL
                  4  4  |  |  |  |  |  | 3#
                         --------+---------(4)LN
                        3  |  |  3  |  |  | 2#
                           |  |      --+------(4)NN
                           |  |     3  3  |  | 2#
                            --------------+------(6)LN
                           3  |           3  |  | 2#
                              |               --+---(8)NN
                              |              2  2  | 1#
                               --------------------+---(11)LN
                              2                    2  | 1#
                                                       --+(19)NN
4  4  4  4  4  4  4  4  3  3  2                       1  1 0
     Huffman code lengths
```

Huffman (cont.)

Symbol	Frequencies	code length	Huffman codes
0	0	-	
.	.	.	
.	.	.	
A	4	3	101
B	1	4	0111
C	0	-	
D	0	-	
E	1	4	0110
F	1	4	0101
G	5	2	11
H	0	-	
I	0	-	
J	1	4	0100
K	2	3	100
L	1	4	0011
M	0	-	
N	0	-	
O	0	-	
P	1	4	0010
Q	1	4	0001
R	0	-	
S	0	-	
T	1	4	0000
.	.	.	
.	.	.	
255	0	-	

Smallest code for each length:	0000	100	11
Number of codes for each length:	8	2	1

Huffman (cont.)

```
nibble
 0000  -4        m-=4;      if(m<1)  m+=23;   )
 0001  -3        m-=3;      if(m<1)  m+=23;   )
 0010  -2        m-=2;      if(m<1)  m+=23;   )
 0011  -1        m-=1;      if(m<1)  m+=23;   )
                                              )  len[s++] = m;
 0100  M  bbbb  m+=bbbb+4; if(m>23) m-=23;   )
 0101  +1        m+=1;      if(m>23) m-=23;   )
 0110  +2        m+=2;      if(m>23) m-=23;   )
 0111  +3        m+=3;      if(m>23) m-=23;   )

 1000  R1     ) use a sequence of these to encode
 1001  R2     ) an integer n = 1,2,3,... then
 1010  R3     )
 1011  R4     )         for(i=1; i<=n; i++) len[s++] = m;

 1100  Z1     ) use a sequence of these to encode
 1101  Z2     ) an integer n = 1,2,3,... then
 1110  Z3     )
 1111  Z4     )         for(i=1; i<=n; i++) len[s++] = 0;
```

Huffman (cont.)

```
zero to  65             Z1   Z4  Z3       (+65)
                         1+4*(4+4*3)      = 65
         65 len[A]=3    +3
         66 len[B]=4    +1
zero to  69             Z2
         69 len[E]=4  )
         70 len[F]=4  ) R2
         71 len[G]=2    -2
zero to  74             Z2
         74 len[J]=4    +2
         75 len[K]=3    -1
         76 len[L]=4    +1
zero to  80             Z3
         80 len[P]=4  )
         81 len[Q]=4  ) R2
zero to  84             Z2
         84 len[T]=4    R1
zero to 256             Z4   Z2   Z2  Z2   (+172)
                         4+4*(2+4*(2+4*2)) = 172

stop
```