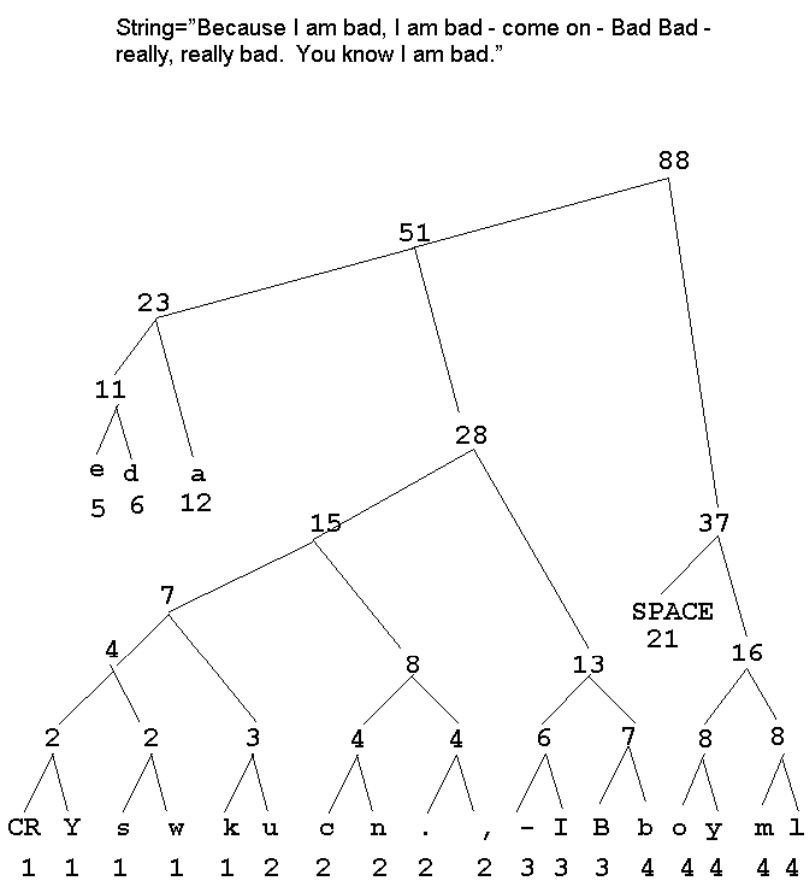
HW2, problem 2.
EECS 207A. UCI. Fall 2004
By Nasser Abbasi

Step 1

First generate the frequency table.
Then use this table to generate the Binary tree. This is the result of these 2 steps:



Frequency Table

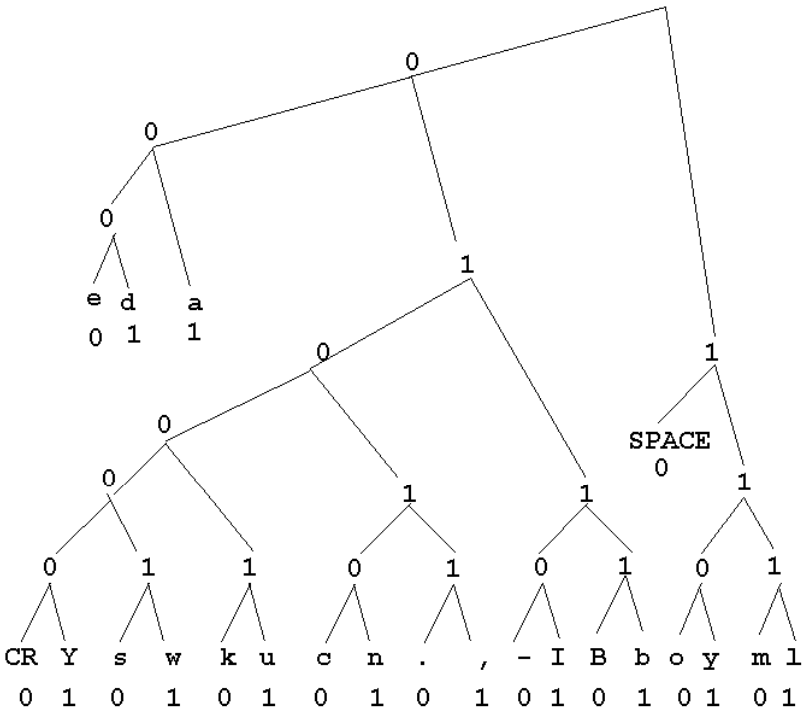
CR	1
w	1
k	1
s	1
Y	1
u	2
c	2
n	2
.	2
,	2
-	3
I	3
B	3
o	4
b	4
y	4
m	4
l	4
e	5
d	6
a	12
SPACE	21

Step 2

Next Generate the Huffman coding for the above tree. This is the result:

String="Because I am bad, I am bad - come on - Bad Bad - really, really bad. You know I am bad."

Code Generation



Frequency Table

CR	1
w	1
k	1
s	1
Y	1
u	2
c	2
n	2
.	2
,	2
-	3
I	3
B	3
o	4
b	4
y	4
m	4
l	4
e	5
d	6
a	12
SPACE	21

Step 3

Next, use the tree to generate the Huffman code and calculate the compression ratio. This is the final result:

Huffman code result

Value	Huffman code	length	frequency	Bit usage
CR	0100001	7	1	7
w	0100011	7	1	7
k	010010	6	1	6
s	0100010	7	1	7
Y	0100001	7	1	7
u	010011	6	2	12
c	010100	6	2	12
n	010101	6	2	12
.	010110	6	2	12
'	010111	6	2	12
-	01100	5	3	15
I	01101	5	3	15
B	01110	5	3	15
o	1100	4	4	16
b	1111	4	4	16
y	1101	4	4	16
m	1110	4	4	16
l	1111	4	4	16
e	0000	4	5	20
d	0001	4	6	24
a	001	3	12	36
SPACE	10	2	21	42

Huffman total bit usage= 341 bits
ASCII bits udage = 87 * 8 = 696 bits
Compression ratio = 341/696 = 48.99 = **49 %**

Conclusion

For the given input, Huffman encoding resulted in 49% saving in data bandwidth.