

Programming Assignment 5

Q1) How many bits were you able to save by using Huffman encoding, compared to a n-bit fixed length code?

Ans) For input 1 (lorem ipsum) huffman encoding saved 148 bits. For input 2 (The adventures of Sherlock Holmes) huffman encoding saved 582162 bits.

Q2) How many characters did you choose to go with – 32, 64, or 128?

Ans) 32

Input - lorem ipsum dolor sit amet, consectetur adipiscing elit. sed et tortor metus. sed at luctus lorem. vivamus faucibus ipsum in diam aliquet, at suscipit augue posuere!

Output:

```
u 000
s 001
o 0100
m 0101
a 0110
v 011100
f 0111010
b 0111011
p 01111
' 100000000
w 1000000010
k 10000000110
j 100000001110
? 10000000111100
y 10000000111101
h 100000001111100
x 100000001111101
z 10000000111111
! 10000001
q 1000001
n 100001
d 10001
l 10010
c 10011
  101
e 1100
r 11010
. 110110
g 1101110
, 1101111
t 1110
i 1111
```

The text was encoded using 677 bits

The text had 165 valid characters
Using a 5-bit fixed length encoding, this would have been 825 long
So we saved 148 bits!

Input - The Adventures of Sherlock Holmes.txt -

<http://www.gutenberg.org/cache/epub/1661/pg1661.txt>

Output:

r 0000
s 0001
e 001
i 0100
h 0101
u 01100
k 0110100
? 011010100
q 0110101010
j 0110101011
z 01101011000
! 01101011001
x 0110101101
' 011010111
, 011011
n 0111
g 100000
f 100001
l 10001
o 1001
a 1010
d 10110
y 101110
v 1011110
b 1011111
t 1100
c 110100
w 110101
m 110110
. 1101110
p 1101111
111

The text was encoded using 2327158 bits
The text had 547386 valid characters
Using a 5-bit fixed length encoding, this would have been 2736930 long
So we saved 409772 bits!