Part I: Huffman Encoding

Given Frequencies

Character	Frequency
e	34
r	22
s	24
t	28
n	15
l	10
i	9
z	8

Huffman Tree Construction

1. Combine $z(8)$ and $i(9)$ to form node $A(17)$.
2. Combine $l(10)$ and $n(15)$ to form node $B(25)$.
3. Combine $A(17)$ and $r(22)$ to form node $C(39)$.
4. Combine $s(24)$ and $B(25)$ to form node $D(49)$.
5. Combine $t(28)$ and $e(34)$ to form node $E(62)$.
6. Combine $C(39)$ and $D(49)$ to form node $F(88)$.
7. Combine $E(62)$ and $F(88)$ to form root $G(150)$.

Huffman Codes

Assigning 0 and 1 down each branch, we get:

Character	Code
t	00
e	01
z	1000
i	1001
r	101
s	110
l	1110
n	1111

Total Bits vs. ASCII

Huffman-Encoded Bits.

$$\begin{aligned}
&e(34) \times 2 = 68, \quad r(22) \times 3 = 66, \quad s(24) \times 3 = 72, \\
&t(28) \times 2 = 56, \quad n(15) \times 4 = 60, \quad l(10) \times 4 = 40, \\
&i(9) \times 4 = 36, \quad z(8) \times 4 = 32. \\
&\text{Total Huffman bits } = 68 + 66 + 72 + 56 + 60 + 40 + 36 + 32 = 430.
\end{aligned}$$

ASCII Bits. Each character would use 8 bits, so for 150 characters:

$$150 \times 8 = 1200 \text{ bits.}$$

Bits Saved.

$$1200 - 430 = 770 \text{ bits saved in total.}$$

Encoding Sample Words

- **"next"**:

$$n = 1111, \quad e = 01, \quad x \text{ (or } z) = 1000, \quad t = 00$$

$$\text{"next"} = 1111\,01\,1000\,00 = 111101100000.$$

- **"stern"**:

$$s = 110, \quad t = 00, \quad e = 01, \quad r = 101, \quad n = 1111$$

$$\text{"stern"} = 110\,00\,01\,101\,1111 = 11000011011111.$$

- **"nertzrents"**:

$$n = 1111,\ e = 01,\ r = 101,\ t = 00,\ z = 1000,\ r = 101,\ e = 01,\ n = 1111,\ t = 00,\ s = 110$$

$$\begin{aligned}
\text{"nertzrents"} &= 1111\,01\,101\,00\,1000\,101\,01\,1111\,00\,110 \\
&= 11110110100100010101111100110.
\end{aligned}$$