Step 6: Last $r=2$ symbols are now encoded with 'r' different code symbols ie '0' and '1' shown Table 4

Table 4:

			Source "Sa"		Source "Sb"	
Source symbol	prob	code	prob	code	prob	code
S_1	0.4		0.4		0.6	← 0
S_2	0.3		0.3		0.4	← 1
S_3	0.2		0.3			
S_4	0.1					

Step 7: As we comming backward '0' may be recomposed as '00' and '01' and '1' may be as '10' and '11' depending on the level of combining. as in Table 5.

Table 5:

			Source "Sa"		Source "Sb"	
Source symbol	prob	code	prob	code	prob	code
S_1	0.4		0.4 ←	1	0.6	← 0
S_2	0.3		0.3 ←	00	0.4	← 1
S_3	0.2		0.3 ←	01		
S_4	0.1					

Step 8 and step 9: This procedure continues till we get codewords corresponding to each source symbol shown in Table 6:

Table 6:

			Source "Sa"		Source "Sb"	
Source symbol	prob	code	prob	code	prob	code
S_1	0.4 ←	1	0.4	1	0.6	0
S_2	0.3 ←	00	0.3	00	0.4	1
S_3	0.2 ←	010	0.3	01		
S_4	0.1 ←	011				

Code Table :

Source symbol	code word	prob	length (binits).
s_1	1	0.4	1
s_2	00	0.3	2
s_3	010	0.2	3
s_4	011	0.1	3

Average length ; $L = \sum_{i=1}^{4} p_i l_i$

$= 0.4 \times 1 + 0.3 \times 2 + 0.2 \times 3 + 0.1 \times 3$

$= 1.9$ binits / msg.

Entropy , $H(s) = -\sum_{i=1}^{4} p_i \log p_i$

$= -[0.4 \log 0.4 + 0.3 \log 0.3 + 0.2 \log 0.2 + 0.1 \log 0.1]$

$= 1.846$ bits / msg.

Code efficiency, $\eta_c = \frac{H(s)}{L} = \frac{1.846}{1.9} \times 100 = 97.15\%$

Code Redundancy, $R_c = 1 - \eta_c = (1 - 0.9715) \times 100$

$= 2.85\%$

Q9. Construct a binary Huffman code by placing the composite symbol "as low as possible".

$s = \{ s_1, s_2, s_3, s_4, s_5, s_6, s_7 \}$ with

$P = \{ 0.4, 0.2, 0.1, 0.1, 0.1, 0.05, 0.05 \}$

Repeate the coding by moving the composite symbol "as high as possible". Compare the variance of the wordlengths and comment the result ?

Sol: 1. Composite symbol placed "as low as possible".

$$q = r + (r-1)\alpha \; ; \; \text{Given } q = 7 \text{ and } r = 2.$$

$$7 = 2 + \alpha \longrightarrow \underline{\underline{\alpha = 5}} \; ; \text{ An integer.}$$

Here we have to proceed '5' stages to reduce the given sources to 2 symbols.

S			Sa		Sb		Sc		Sd		Se	
Source	p	code	p	code	p	code	p	code	p	code	p	code
s_1	0.4		0.4		0.4		0.4		0.4		0.6	0
s_2	0.2		0.2		0.2		0.2		0.4	0	0.4	1
s_3	0.1		0.1		0.2		0.2	0	0.2	1		
s_4	0.1		0.1		0.1	0	0.2	1				
s_5	0.1		0.1	0	0.1	1						
s_6	0.05	0	0.1	1								
s_7	0.05	1										

<u>Code Table:</u>

Source symbol	prob	r. code	code	length
s_1	0.4	1	1	1
s_2	0.2	10	01	2
s_3	0.1	0100	0010	4
s_4	0.1	1100	0011	4
s_5	0.1	0000	0000	4
s_6	0.05	01000	00010	5
s_7	0.05	11000	00011	5

Avg lgf; $L = \sum_{i=1}^{7} p_i l_i$

$= 0.4\times1 + 0.2\times2 + 0.1\times4\times3 + 0.05\times5\times2$

$= \underline{\underline{2.5 \text{ binits/msg}}}$

$H(S) = -\sum_{i=1}^{7} p_i \log p_i$

$= \underline{\underline{2.4219 \text{ bits/msg}}}$

$$\eta_c = \frac{H(S)}{L} = \frac{2.4219}{2.5} \times 100 = \underline{\underline{96.88\%}}$$

Variance, $Var(X) = E[(X-\mu)^2]$ where $\mu \to$ average value.

Then the variance of the word length;

$$\boxed{Var(l_i) = \sum_{i=1}^{7} p_i (l_i - L)^2}$$

$$= 0.4(1-2.5)^2 + 0.2(2-2.5)^2 + 0.1(4-2.5)^2 \times 3$$

$$+ 0.05(5-2.5)^2 \times 2$$

$$= \underline{\underline{2.25}}$$

2. Composite symbol placed "as high as possible".

S			Sa		Sb		Sc		Sd		Se	
Source	p	code	p	code	p	code	p	code	p	code	p	code
s_1	0.4		0.4		0.4		0.4		0.4		0.6	0
s_2	0.2		0.2		0.2		0.2		0.4	0	0.4	1
s_3	0.1		0.1		0.2		0.2	0	0.2	1		
s_4	0.1		0.1		0.1	0	0.2	1				
s_5	0.1		0.1	0	0.1	1						
s_6	0.05	0	0.1	1								
s_7	0.05	1										

<u>Code Table</u> :

Source symbol	prob	r. Codeword	Codeword	length
s_1	0.4	00	00	2
s_2	0.2	11	11	2
s_3	0.1	110	011	3
s_4	0.1	001	100	3
s_5	0.1	101	101	3
s_6	0.05	0010	0100	4
s_7	0.05	1010	0101	4

$L = \sum_{i=1}^{7} p_i l_i$

$= 0.4 \times 2 + 0.2 \times 2 + 0.1 \times 3 \times 3 + 0.05 \times 4 \times 2$

$= \underline{\underline{2.5 \text{ binits/msg}}}$

$H(S) = \underline{\underline{2.4219 \text{ bits/msg}}}$

$$\eta_c = \frac{H(s)}{L} = \frac{2.4219}{2.5} \times 100 = \underline{96.88\%}$$

$$Var(l_i) = \sum_{i=1}^{7} p_i (l_i - L)^2$$

$$= 0.4(2-2.5)^2 + 0.2(2-2.5)^2 + 0.1(3-2.5)^2 \times 3$$

$$+ 0.05(4-2.5)^2 \times 2$$

$$= \underline{0.45}$$

When the composite symbol is moved as high as possible, the variance of the word length over source symbols would become smaller, which is desirable.

HW 9. Consider a source with 8 alphabets A to H with ~~represent~~ respective probabilities of 0.22, 0.20, 0.18, 0.15, 0.10, 0.08, 0.05, 0.02.

(a) Construct a binary compact code using Huffman coding and determine the code efficiency.

(b) Construct a ternary compact code using huffman coding and determine the code efficiency.

(c) Construct a quarternary compact code and determine the code efficiency.

Ans. (a) Binary code.

Code table:

Msg	prob	Codeword	Length
A	0.22	10	2
B	0.20	11	2
C	0.18	000	3
D	0.15	001	3
E	0.10	011	3
F	0.08	0100	4
G	0.05	01010	5
H	0.02	01011	5

$L = \underline{2.8}$ binits/msg

$H(s) = \underline{2.7535}$ bits/msg

$\eta_c = \underline{98.34\%}$

(b) Huffman Ternary Code :

Consider $q = r + (r-1)\alpha$

where $q = 8$; $r = 3$.

Substitute $r \rightarrow$ $q = 3 + 2\alpha$

$$\alpha = \frac{q-3}{2}.$$

Then q should have the value 5, 7, 9, 11 ... to 'α' be an integer.

if $q = 8$; $\alpha = \frac{8-3}{2} = \frac{5}{2} = \underline{2.5}$; not an integer

if $q = 9$; $\alpha = \frac{9-3}{2} = \underline{3}.$

So add "one dummy symbol" 'I' with zero probabilities. So $q = 9$; with code alphabet $x = \{0, 1, 2\}$.

Source "S"			"Sa"		"Sb"		"Sc"	
Source	prob	code	prob	code	prob	code	prob	code.
A	0.22		0.22		0.25		0.53	0
B	0.2		0.2		0.22		0.25	1
C	0.18		0.18		0.2	0	0.22	2
D	0.15		0.15		0.18	1		
E	0.1		0.1	0	0.15	2		
F	0.08		0.08	1				
G	0.05	0	0.07	2				
H	0.02	1						
Dummy Symbol I	0	2	Discard					

Code Table :

Msg	prob	r.code	code	length
A	0.22	2	2	1
B	0.2	00	00	2
C	0.18	10	01	2
D	0.15	20	02	2
E	0.10	01	10	2
F	0.08	11	11	2
G	0.05	021	120	3
H	0.02	121	121	3

$L^{(3)} = 1.85$ trinits/msg

$H(S) = 2.7585$ bits/msg

$H_3(S) = \dfrac{H(S)}{\log 3} = 1.7373$ ternary unit/msg

$\eta_c^{(3)} = \dfrac{H_3(S)}{L^{(3)}} = \dfrac{1.7373}{1.85}$

$= 93.91\ \%$

(c) Huffman Quarternary Code

Here $q = 8$ and $r = 4$.

We know that $q = r + (r-1)\alpha$

$q = 4 + 3\alpha$

$\alpha = \dfrac{q-4}{3}$;

q should be $7, 10, 13, 16 \ldots\ldots$

Given that $q = 8$, So take new $q = 10$ by adding two "dummy symbols" I and J each with probabilities zero. if $q = 10$; $\alpha = \dfrac{10-4}{3} = 2$, An integer.

with code alphabet $X = \{0, 1, 2, 3\}$.

Source "S"			"Sa"		"Sb"	
Source	prob	code	prob	code	prob	code
A	0.22		0.22		0.4	0
B	0.2		0.2		0.22	1
C	0.18		0.18		0.2	2
D	0.15		0.15	0	0.18	3
E	0.1		0.1	1		
F	0.08		0.08	2		
G	0.05	0	0.07	3		
H	0.02	1				
I	0	2				
J	0	3				

Dummy symbols

Discard

Code Table

Msg	prob	codeword	codeword	lgt
A	0.22	1	1	1
B	0.2	2	2	1
C	0.18	3	3	1
D	0.15	00	00	2
E	0.10	10	01	2
F	0.08	210	012	2
G	0.05	030	030	3
H	0.02	130	031	3

$L^{(4)} = 1.47$ quarternary digits / msg

$H(s) = 2.7535$ bits/msg.

$H_4(s) = \frac{H(s)}{\log 4} = 1.37675$ quarternary units/msg

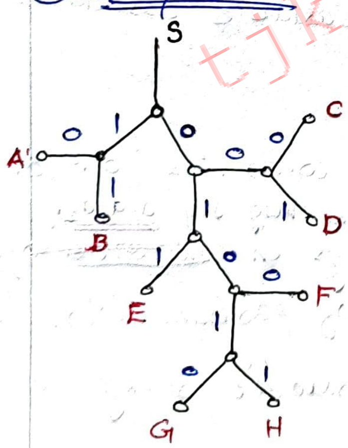
$\eta_c^{(4)} = \frac{H_4(s)}{L^{(4)}} = 93.66\%$

Comparison

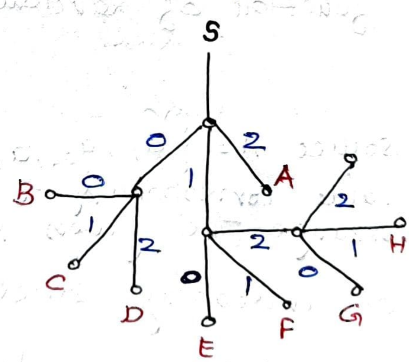
Type of coding	Code efficiency
Binary coding	98.34%
Ternary coding	93.91%
Quarternary coding	93.66%

Code tree

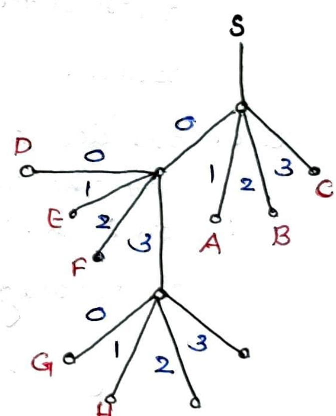
ⓐ Binary code



ⓑ ternary code



ⓒ Quarternary Code



HW10: Apply Huffman encoding procedure for the following set of msgs and determine the η_c of the binary code formed.

x_1	x_2	x_3
0.7	0.15	0.15

Also apply Huffman coding to the second order extension for the above msg and calculate how much the efficiency be improved?