Table 1:

Source symbol	prob	code	Source "Sa" prob	code
s_1	0.4		0.4	
s_2	0.3		0.3	
s_3	0.2		0.3	
s_4	0.1			

Two choises for placing the "composite symbol". One way is placing the "composite symbol" "as low as possible." as shown in Table 1. The second way is placing the composite symbol "as high as possible" as shown in Table 2.

Table 2:

Source symbol	prob	code	Source "Sa" prob	code
s_1	0.4		0.4	
s_2	0.3		0.3	
s_3	0.2		0.3	
s_4	0.1			

Step 4 and steps: Follow the first method to form reduced source "Sb" by combining last "r = 2" symbols of "Sa".

Table 3:

Source symbol	prob	code	Source "Sa" prob	code	Source "Sb" prob	code
s_1	0.4		0.4		0.6	
s_2	0.3		0.3		0.4	
s_3	0.2		0.3			
s_4	0.1					

Follow the method by placing the composite symbol "as low as possible".

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

Table 4:

Source symbol	prob	code	Source "Sa" prob	Source "Sa" code	Source "Sb" prob	Source "Sb" 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 symbol	prob	code	Source "Sa" prob	Source "Sa" code	Source "Sb" prob	Source "Sb" 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 symbol	prob	code	Source "Sa" prob	Source "Sa" code	Source "Sb" prob	Source "Sb" 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 ?