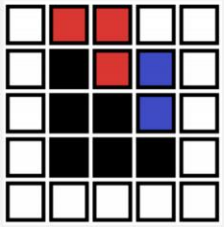
HUFFMAN CODING

• **Exercise** Construct a Huffman code for this image, determine the average bits per pixel, the compression ratio and corresponding relative redundancy given by your code.



มีค่าเท่ากับ 0
มีค่าเท่ากับ 2
มีค่าเท่ากับ 4
มีค่าเท่ากับ 7

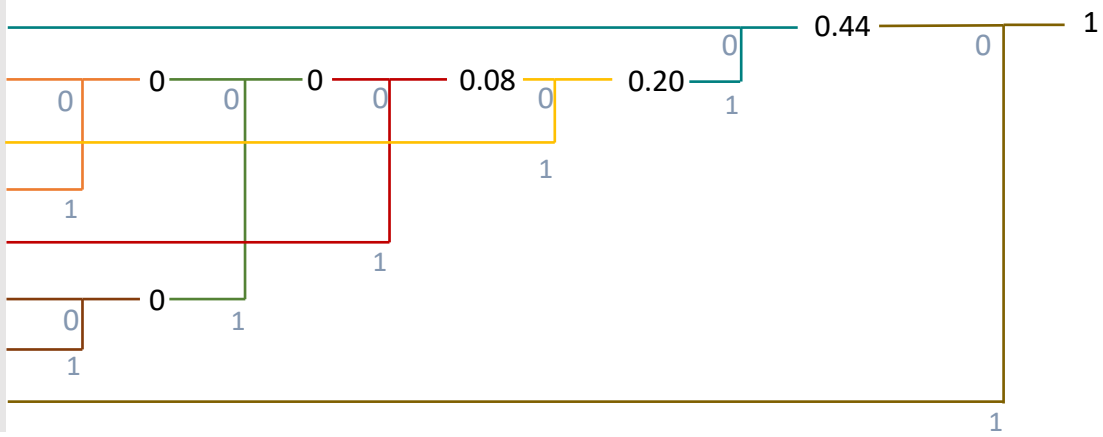
Grey Value	0	1	2	3	4	5	6	7
P	0.24	0	0.12	0	0.08	0	0	0.56

สมาชิกในกลุ่ม

นวฤทธิ์ หล่องคุ้ม 60070501028
สุกษมา จิตอักษร 60070501065
ธนพร ปิติอนุสรณ์ 60070501071
วรรษมน พานทอง 60070501083

Gray value	Prob(a)
0	0.24
1	0
2	0.12
3	0
4	0.08
5	0
6	0
7	0.56

กำหนดให้ เส้นบนมีค่าเท่ากับ 0 และ เส้นด่านล่างมีค่าเท่ากับ 1



Gray value	Huffman code
0	00
1	010000
2	011
3	010001
4	0101
5	010010
6	010011
7	1

Determine the average bits per pixel

(0.24*2)+(0*6)+(0.12*3)+(0*6)+(0.08*4)+(0*6)+(0*6)+(0.56*1)=1.72

The compression ratio and corresponding relative redundancy given by your code

Compression ratio = (5*5*3)/(5*5*1.72)=1.74

Redundancy ratio = 1-(1/1.74) = 0.4253

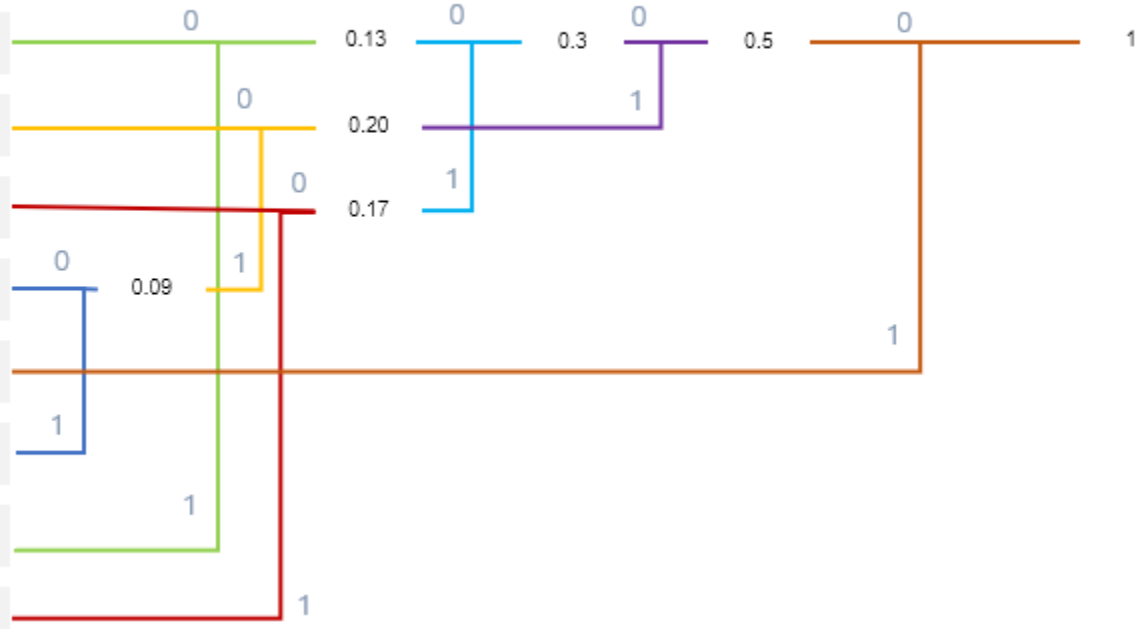
In-class Exercise

1. Construct a Huffman code for each of the probability tables given:

gray value	0	1	2	3	4	5	6	7
probability (a)	0.07	0.11	0.08	0.04	0.5	0.05	0.06	0.09
probability (b)	0.13	0.12	0.13	0.13	0.12	0.12	0.12	0.13
probability (c)	0.09	0.13	0.15	0.1	0.14	0.12	0.11	0.16

In each case determine the average bits per pixel given by your code.

Gray value	Prob(a)
0	0.07
1	0.11
2	0.08
3	0.04
4	0.5
5	0.05
6	0.06
7	0.09



Gray value	Code	L
0	0000	4
1	010	3
2	0010	4
3	0110	4
4	1	1
5	0111	4
6	0001	4
7	0011	4

Determine the average bits per pixel

(0.07*4)+(0.11*3)+(0.08*4)+(0.04*4)+(0.5*1)+(0.05*4)+(0.06*4)+(0.09*4) = 2.39 bits/pixel

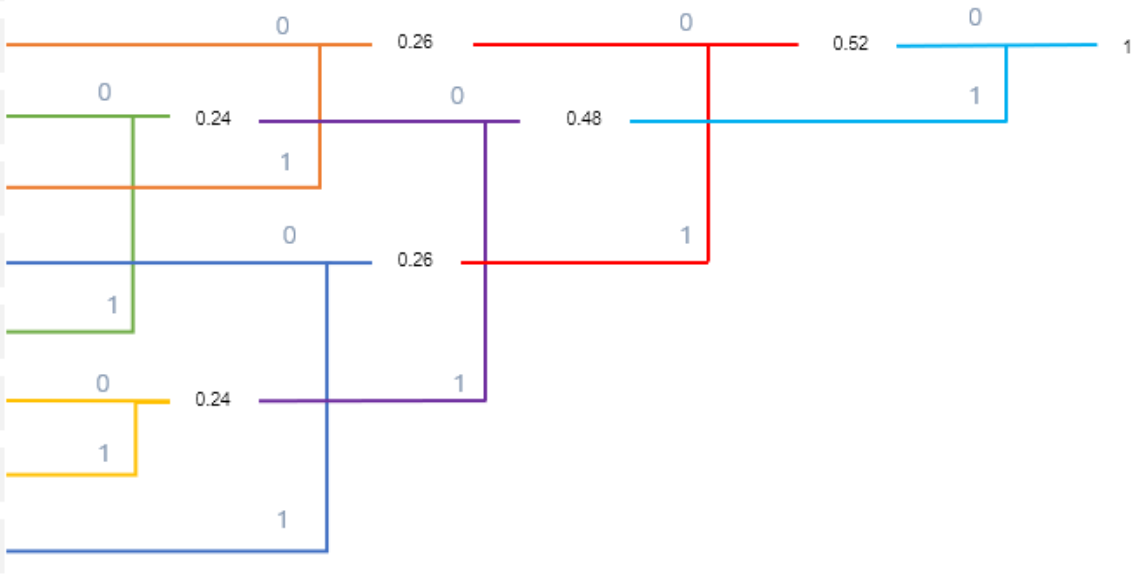
In-class Exercise

1. Construct a Huffman code for each of the probability tables given:

gray value	0	1	2	3	4	5	6	7
probability (a)	0.07	0.11	0.08	0.04	0.5	0.05	0.06	0.09
probability (b)	0.13	0.12	0.13	0.13	0.12	0.12	0.12	0.13
probability (c)	0.09	0.13	0.15	0.1	0.14	0.12	0.11	0.16

In each case determine the average bits per pixel given by your code.

Gray value	Prob(b)
0	0.13
1	0.12
2	0.13
3	0.13
4	0.12
5	0.12
6	0.12
7	0.13



Gray value	Code	L
0	000	3
1	100	3
2	001	3
3	010	3
4	101	3
5	110	3
6	111	3
7	011	3

Determine the average bits per pixel

(0.13*3)+(0.12*3)+(0.13*3)+(0.13*3)+(0.12*3)+(0.12*3)+(0.12*3)+(0.13*3) = 3 bits/pixel

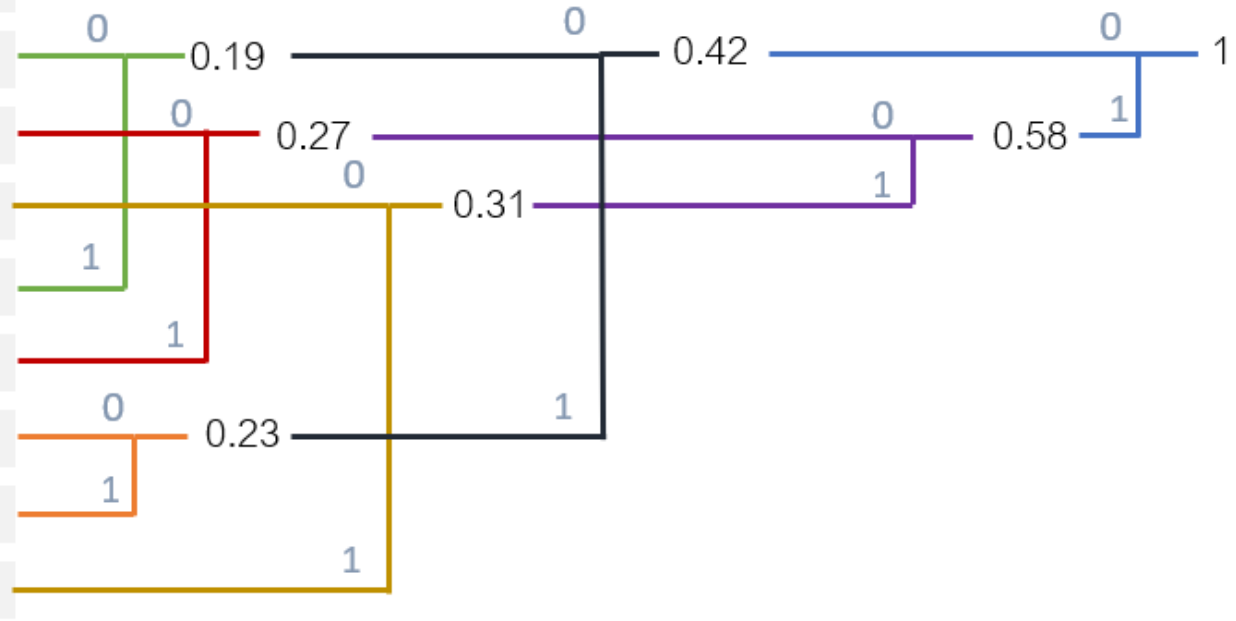
In-class Exercise

1. Construct a Huffman code for each of the probability tables given:

gray value	0	1	2	3	4	5	6	7
probability (a)	0.07	0.11	0.08	0.04	0.5	0.05	0.06	0.09
probability (b)	0.13	0.12	0.13	0.13	0.12	0.12	0.12	0.13
probability (c)	0.09	0.13	0.15	0.1	0.14	0.12	0.11	0.16

In each case determine the average bits per pixel given by your code.

Gray value	Prob(c)
0	0.09
1	0.13
2	0.15
3	0.1
4	0.14
5	0.12
6	0.11
7	0.16



Gray value	Code	L
0	000	3
1	100	3
2	110	3
3	001	3
4	101	3
5	010	3
6	011	3
7	111	3

Determine the average bits per pixel

(0.09*3)+(0.13*3)+(0.15*3)+(0.1*3)+(0.14*3)+(0.12*3)+(0.11*3)+(0.16*3) = 3 bits/pixel

2. Encode each of the following binary images using RLE:

(a)					
1	0	0	1	1	1
0	1	0	1	1	1
1	0	0	1	1	1
0	1	1	1	0	1
1	0	1	0	1	1
0	1	1	1	1	0

(b)					
1	0	1	0	0	0
0	0	1	1	0	1
1	1	0	0	0	0
0	0	0	0	1	1
1	1	1	1	0	0
1	1	1	0	0	0

**RLE encoding with starting number 0s

(a)						
1	0	0	1	1	1	(0123)
0	1	0	1	1	1	(1113)
1	0	0	1	1	1	(0123)
0	1	1	1	0	1	(1311)
1	0	1	0	1	1	(011112)
0	1	1	1	1	0	(141)

(b)						
1	0	1	0	0	0	(0113)
0	0	1	1	0	1	(2211)
1	1	0	0	0	0	(024)
0	0	0	0	1	1	(42)
1	1	1	1	0	0	(042)
1	1	1	0	0	0	(033)

3. Using RLE, encode the following 4-bit image:

1	1	3	3	1	1
1	7	10	10	7	1
6	13	15	15	13	6
6	13	15	15	13	6
1	7	10	10	7	1
1	1	3	3	1	1

Encoding 4-bit image with gray code

gray value	0	1	2	3	4	5	6	7
Gray code	0000	0001	0011	0010	0110	0111	0101	0100
gray value	8	9	10	11	12	13	14	15
Gray code	1100	1101	1111	1110	1010	1011	1001	1000

**RLE encoding with starting number 0s

0001	0001	0010	0010	0001	0001	
0001	0100	1111	1111	0100	0001	
0101	1011	1000	1000	1011	0101	
0101	1011	1000	1000	1011	0101	
0001	0100	1111	1111	0100	0001	
0001	0001	0010	0010	0001	0001	

1	1	0	0	1	1	(0222)
1	0	1	1	0	1	(011211)
1	1	0	0	1	1	(0222)
1	1	0	0	1	1	(0222)
1	0	1	1	0	1	(011211)
1	1	0	0	1	1	(0222)

0^{th} plane

0	0	1	1	0	0	(222)
0	0	1	1	0	0	(222)
0	1	0	0	1	0	(11212)
0	1	0	0	1	0	(11212)
0	0	1	1	0	0	(222)
0	0	1	1	0	0	(222)

1^{st} plane

0	0	0	0	0	0	(6)
0	1	1	1	1	0	(141)
1	0	0	0	0	1	(0141)
1	0	0	0	0	1	(0141)
0	1	1	1	1	0	(141)
0	0	0	0	0	0	(6)

2^{nd} plane

0	0	0	0	0	0	(6)
0	0	1	1	0	0	(222)
0	1	1	1	1	0	(141)
0	1	1	1	1	0	(141)
0	0	1	1	0	0	(222)
0	0	0	0	0	0	(6)

3^{rd} plane

4. Apply JPEG compression to an 8x8 block consisting of

A. All the same value.

B. The left half one value, and the right half another.

C. Random values in the range 0-255 range.

Compare the length of the code vector in each case.

```
b = double(block) - 128;
bd = dct2(b);
q = [16 11 10 16 24 40 51 61;...
    12 12 14 19 26 58 60 55;...
    14 13 16 24 40 57 69 56;...
    14 17 22 29 51 87 80 62;...
    18 22 37 56 68 109 103 77;...
    24 35 55 64 81 104 113 92;...
    49 64 78 87 103 121 120 101;...
    72 92 95 98 112 100 103 99];
bq = round(bd./q);
```

```
block = [125 125 125 125 125 125 125 125;...
    125 125 125 125 125 125 125 125;...
    125 125 125 125 125 125 125 125;...
    125 125 125 125 125 125 125 125;...
    125 125 125 125 125 125 125 125;...
    125 125 125 125 125 125 125 125;...
    125 125 125 125 125 125 125 125;...
    125 125 125 125 125 125 125 125];
```

	1	2	3	4	5	6	7	8
1	-1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0

A. Length = 1

```
block = [125 125 125 125 0 0 0 0;...
    125 125 125 125 0 0 0 0;...
    125 125 125 125 0 0 0 0;...
    125 125 125 125 0 0 0 0;...
    125 125 125 125 0 0 0 0;...
    125 125 125 125 0 0 0 0;...
    125 125 125 125 0 0 0 0;...
    125 125 125 125 0 0 0 0];
```

	1	2	3	4	5	6	7	8
1	-33	41	0	-10	0	3	0	1
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0

B. Length = 29

```
block = [14 12 125 95 0 90 36 0;...
    12 155 18 99 0 70 44 77;...
    125 125 136 125 50 0 7 0;...
    75 115 25 125 0 74 33 0;...
    25 125 66 15 0 20 0 30;...
    125 5 12 125 0 0 14 44;...
    105 125 95 88 0 10 0 0;...
    12 75 1 11 0 65 0 70];
```

	1	2	3	4	5	6	7	8
1	-40	16	0	-4	-1	1	0	-2
2	5	-1	-5	-1	-2	0	0	0
3	-1	-6	-1	0	-1	0	1	0
4	-3	-2	0	0	0	1	0	0
5	-3	-3	0	0	-1	0	0	0
6	2	2	-1	0	0	1	0	0
7	0	0	0	0	1	1	0	0
8	0	1	0	0	0	0	0	1

C. Length = 64

5. Write a Matlab code to compress the grayscale image (.bmp). Using JPEG compression to compress this image using greater and greater compression rates. What is the largest quantisation scale factor for which the image is still recognizable? How many 0s are in the DCT block matrix?

	1	2	3	4	5	6	7	8
1	-32	1	0	0	0	0	0	0
2	-5	1	0	0	0	0	0	0
3	-4	-1	0	0	0	0	0	0
4	1	1	0	0	0	0	0	0
5	-1	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0

Fig1 show Quantization by q

	1	2	3	4	5	6	7	8
1	-16	1	0	0	0	0	0	0
2	-2	0	0	0	0	0	0	0
3	-2	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0

Fig2 show Quantization by 2*q

	1	2	3	4	5	6	7	8
1	-11	0	0	0	0	0	0	0
2	-2	0	0	0	0	0	0	0
3	-1	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0

Fig3 show Quantization by 3*q

	1	2	3	4	5	6	7	8
1	-8	0	0	0	0	0	0	0
2	-1	0	0	0	0	0	0	0
3	-1	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0

Fig4 show Quantization by 4*q

	1	2	3	4	5	6	7	8
1	-65	2	0	0	0	0	0	0
2	-10	2	1	0	0	0	0	0
3	-8	-1	-1	0	0	0	0	0
4	1	1	0	0	0	0	0	0
5	-1	0	0	0	0	0	0	0
6	0	-1	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0

Fig5 show Quantization by 0.5*q

	1	2	3	4	5	6	7	8
1	-22	1	0	0	0	0	0	0
2	-3	1	0	0	0	0	0	0
3	-3	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0

Fig6 show Quantization by 1.5*q

จากการเพิ่ม scale quantisation เมื่อเพิ่มขึ้น จะทำให้มีจำนวน 0 ต่อท้ายมากขึ้น แต่เมื่อเพิ่มได้ระยะหนึ่ง ความยาว 0 ที่ต่อท้ายจะคงที่

```
img = rgb2gray(imread('sample_640×426.bmp'));

x = 255;
y = 120;
block = img(x:x+7, y:y+7);
b = double(block) - 128;
bd = dct2(b);
q = [16 11 10 16 24 40 51 61;...
    12 12 14 19 26 58 60 55;...
    14 13 16 24 40 57 69 56;...
    14 17 22 29 51 87 80 62;...
    18 22 37 56 68 109 103 77;...
    24 35 55 64 81 104 113 92;...
    49 64 78 87 103 121 120 101;...
    72 92 95 98 112 100 103 99];
bp = round(bd./q);
bp2 = round(bd./(2*q));
bp3 = round(bd./(3*q));
bp4 = round(bd./(4*q));
bphalf = round(bd./(0.5*q));
bp15 = round(bd./(1.5*q));
```