

Final 15/16 - 1st

Exercise 1:
~~~~~

1. c / 2. d / 3. a / 4.  
5. a,d / 6. b / 7. d / 8. a

Exercise 2:  
~~~~~

1. T / 2. F / 3. T / 4. F / 5. F / 6. F
7. T / 8. F / 9. F / 10. T / 11. F / 12. T

Exercise 3:
~~~~~

$$\begin{array}{lll} a \rightarrow 3 & b \rightarrow 6 & c \rightarrow 1 \\ d \rightarrow 2 & e \rightarrow 5 & f \rightarrow 4 \end{array}$$

## Exercise 4:

| a) Level | count | pdf     | cdf     | round $((\text{Level}-1) * \text{cdf})$ |
|----------|-------|---------|---------|-----------------------------------------|
| 0        | 34    | 0.00207 | 0.00207 | 0                                       |
| 1        | 50    | 0.00305 | 0.00512 | 0                                       |
| 2        | 500   | 0.03052 | 0.03564 | 0                                       |
| 3        | 1500  | 0.9155  | 0.12719 | 1                                       |
| 4        | 2700  | 0.16479 | 0.29198 | 2                                       |
| 5        | 4500  | 0.27465 | 0.56663 | 4                                       |
| 6        | 4000  | 0.24414 | 0.81077 | 6                                       |
| 7        | 3100  | 0.18921 | 1       | 7                                       |
| Total    | 16384 |         |         |                                         |

Level 0: 584

Level 1: 1500

b) it will remain the same

Level 2: 2700

Level 3: 0

Level 4: 4500

Level 5: 0

Level 6: 4000

Level 7: 3100

## Exercise 5:

a)

|    |   |    |    |    |
|----|---|----|----|----|
| 13 | 0 | 13 | 6  | 8  |
| 0  | 0 | 4  | 7  | 9  |
| 14 | 0 | 7  | 3  | 12 |
| 0  | 9 | 9  | 6  | 1  |
| 8  | 5 | 15 | 11 | 4  |

|    |    |    |
|----|----|----|
| -1 | -2 | -1 |
| 0  | 0  | 0  |
| 1  | 2  | 1  |

Horizontal

|    |   |   |
|----|---|---|
| -1 | 0 | 1 |
| -2 | 0 | 2 |
| -1 | 0 | 1 |

Vertical

$$\text{Horizontal} : (0 \times (-1)) + (4 \times (-2)) + \dots = 18$$

vertical : 10

$$\text{Magnitude} = 10 + 18 = 28$$

b) 0 0 3 4 6 7 7 9 9

We replace 7 with 6

c) غير مطلوب

```

def apply_median_filter(image, kernel_size):
    rows, cols = image.shape
    pad_width = kernel_size // 2

    padded_image = np.pad(image, pad_width, mode='edge')
    output_image = np.zeros_like(image)

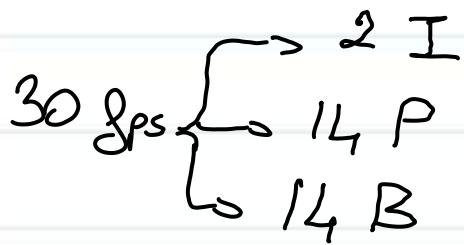
    for i in range(rows):
        for j in range(cols):
            window = padded_image[i:i+kernel_size, j:j+kernel_size]
            median_value = np.median(window)
            output_image[i, j] = median_value

    return output_image

```

Exercise 6:

H.263 uses 4:2:1 subsampling, so 16 bits per pixel.



$$\text{frame size} = 704 \times 576 \times 16 = 6488064 \text{ bits}$$

$$\hookrightarrow \left( 2 \times \frac{5s}{10} + 14 \times \frac{5s}{20} + 14 \times \frac{5s}{40} \right) \rightarrow \text{size for } \underline{1 \text{ sec}}$$

↳ 10:1      ↳ twice as ↓      ↳ twice as ↑

$$2110080 \times 90 \times 60 = 43794432000 \text{ bits} \approx 5.098 \text{ GBbytes}$$

Exercise 7:

// auto-mayhem