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:

a → 3 b → 6 c → 1
d → 2 e → 5 f → 4

Exercise 4:

a)

Level	count	pdf	cdf	round((Level-1)*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

Level 2: 2700

Level 3: 0

Level 4: 4500

Level 5: 0

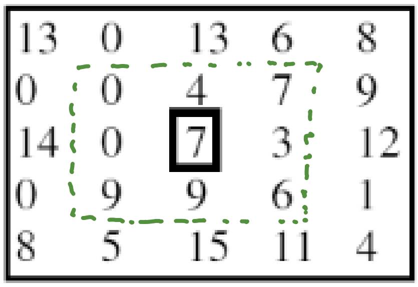
Level 6: 4000

Level 7: 3100

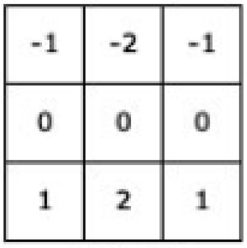
b) it will remain the same

Excercise 5:

a)

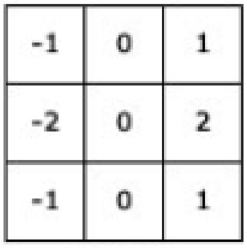


13	0	13	6	8
0	0	4	7	9
14	0	7	3	12
0	9	9	6	1
8	5	15	11	4



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

Horizontal



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

Vertical

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

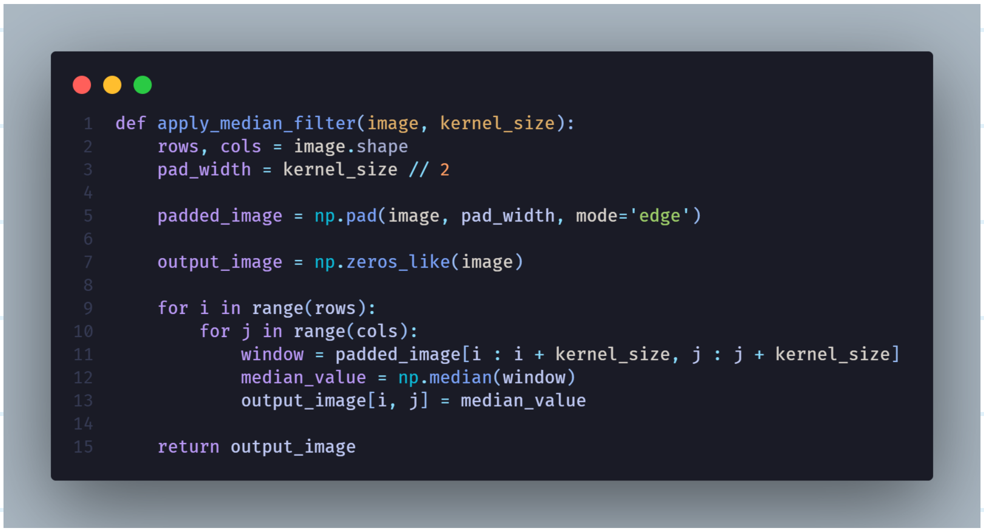
vertical: 10

Magnitude = $10 + 18 = 28$

b) $0 \; 0 \; 3 \; 4 \; \underbrace{6} \; 7 \; 7 \; 9 \; 9$

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 4:2:2 subsampling, so 16 bits per pixel.

30 fps → 2 I
→ 14 P
→ 14 B

frame size = 704 × 576 × 16 = 6 488 064 bits

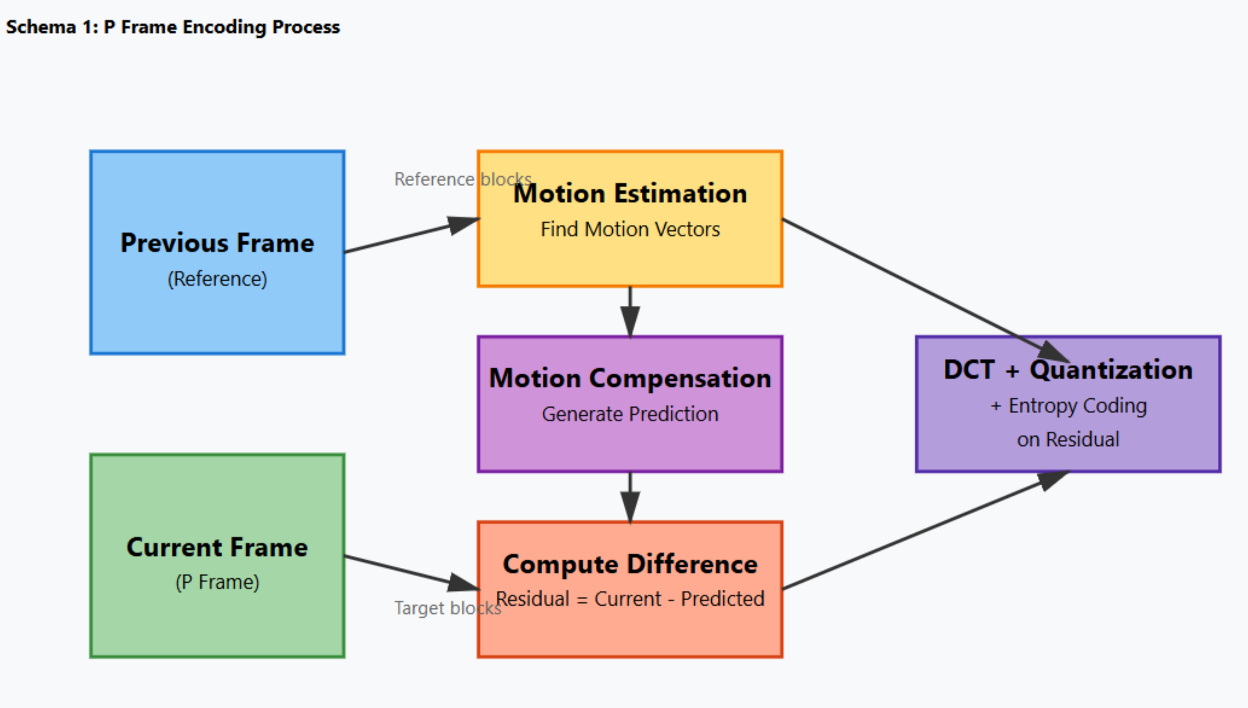
$$\hookrightarrow \left(2 \times \frac{fs}{10} + 14 \times \frac{fs}{20} + 14 \times \frac{fs}{40}\right) \rightarrow \text{size for 1 sec}$$

↳ 10:1 ↳ twice as I ↳ twice as P

8110080 × 90 × 60 = 43794432000 bits ≈ 5.098 GBytes

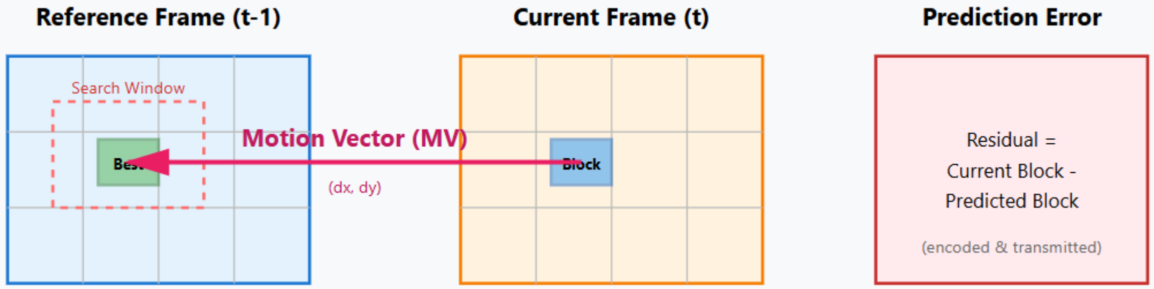
Exercise 7:

a)



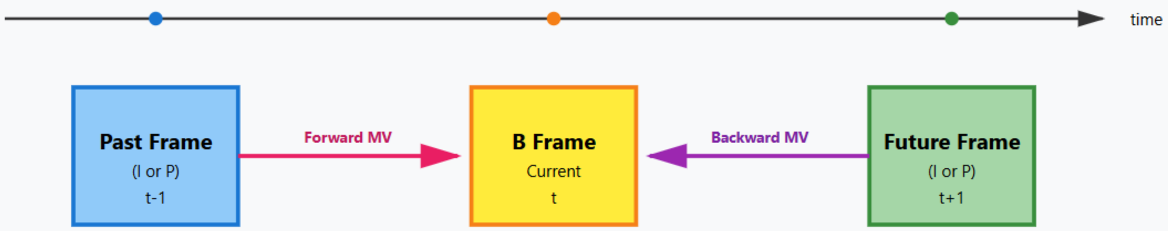
Schema 1: P Frame Encoding Process

Schema 2: Motion Compensation & Estimation Process (Detailed)

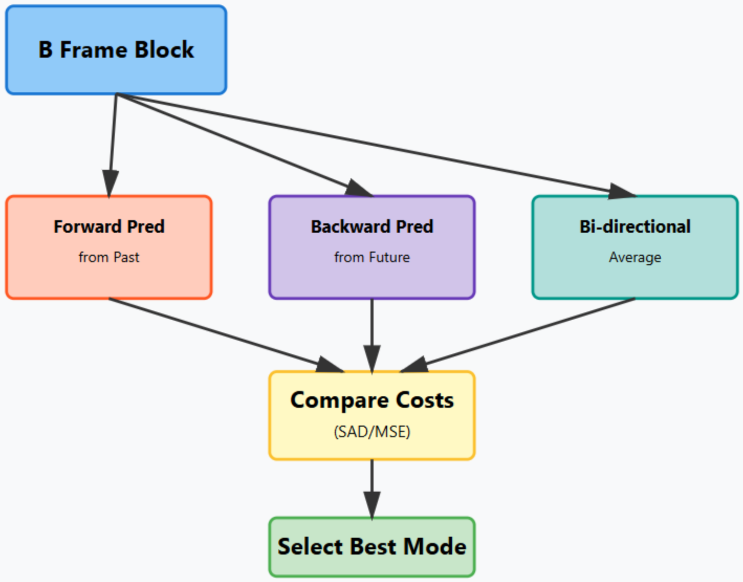


b)

Schema 3: B Frame Encoding with Bidirectional Prediction

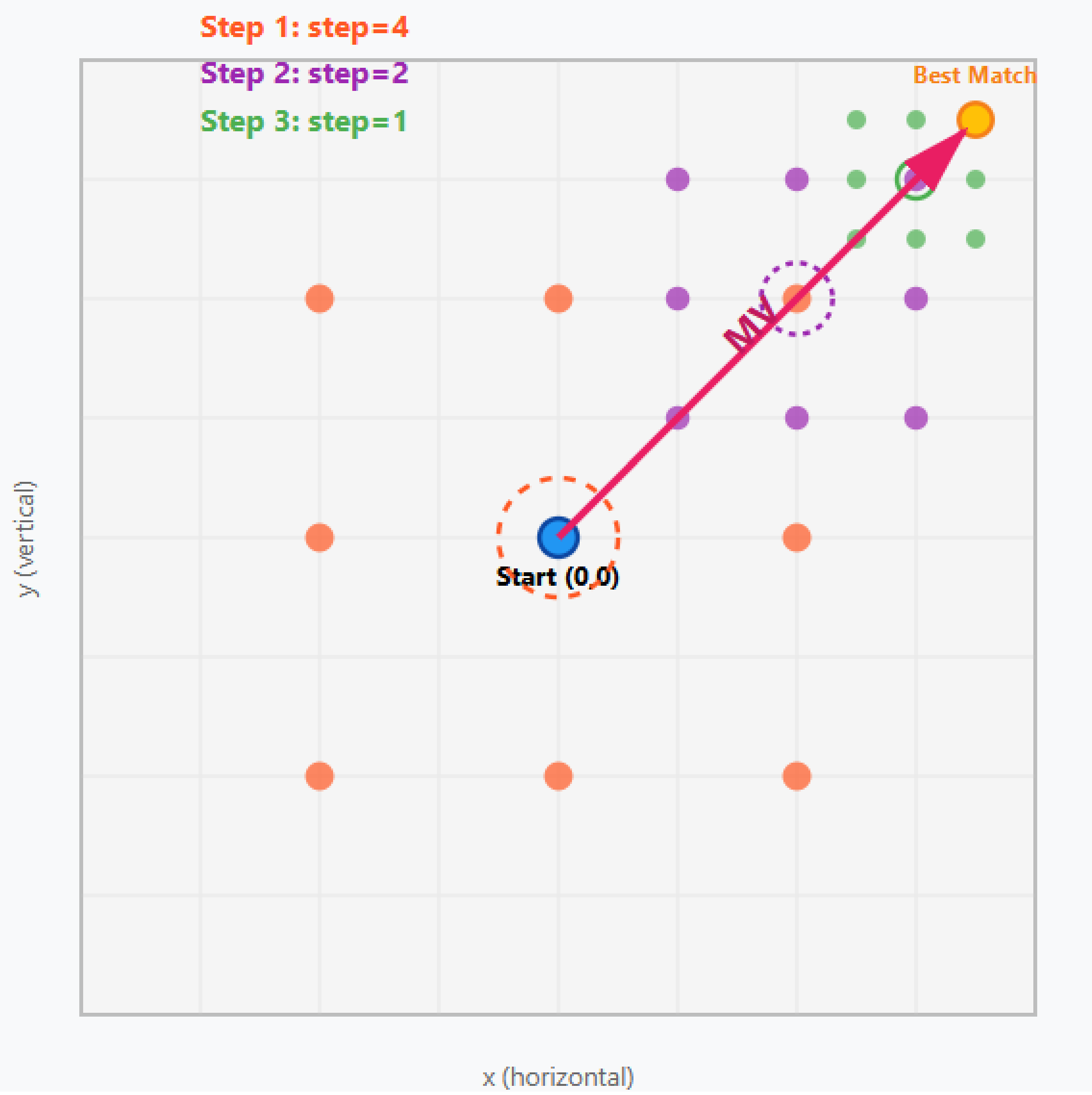


Schema 4: Adapted Motion Compensation for B Frames



c)

Schema 5: 2D Logarithmic Search Algorithm


Step 1: step=4
Step 2: step=2
Step 3: step=1
Best Match
MV
Start (0,0)
y (vertical)
x (horizontal)