Homework #4 – Entropy Coding

314552036　周子翔

I. Introduction

This assignment implements a simplified JPEG-style image compression pipeline using run-length encoding (RLE) and decoding on block-based DCT coefficients. The goal is to understand how quantization, zigzag scanning, and RLE reduce redundancy in the frequency domain, and to compare compression performance under two different quantization tables.
The main steps include:

- Converting the input *lena.png* to grayscale and dividing it into 8×8 blocks.
- Applying the 2D-DCT to each block.
- Quantizing the DCT coefficients using two different quantization tables.
- Applying zigzag scan + run-length encoding.
- Running the decoding pipeline: RLE decoding → inverse zigzag → dequantization → IDCT.
- Comparing the reconstructed image quality and encoded sizes.

II. Method

- **Preprocessing**
  The input *lena.png* is converted to grayscale and padded (if needed) so that both dimensions are multiples of 8. The image is then split into non-overlapping 8×8 blocks using a raster-scan order.
- **2D Discrete Cosine Transform**
  For each 8×8 block, the 2D-DCT is computed using:

$$C = D X D^T$$

  where X is the input block and D is the DCT transform matrix.
  The transform concentrates energy into low-frequency coefficients, making quantization more effective.
- **Quantization**
  Two quantization tables, **Q1** and **Q2**, are provided.
  Each DCT coefficient is quantized by:

$$\widehat{C}(i,j) = round(\frac{C(i,j)}{Q(i,j)})$$

A larger quantization value → more aggressive compression → more zero coefficients → shorter RLE sequences.

- **Zigzag Scan**
  Within each block, quantized DCT coefficients are visited in **zigzag order**, which groups low-frequency coefficients earlier and high-frequency ones later (which are mostly zero after quantization).
  This increases the run-lengths for zeros, improving RLE performance.
- **Run-Length Encoding (RLE)**
  RLE encodes each block as a sequence of (RUNLENGTH, VALUE) pairs.

  RUNLENGTH = number of consecutive zeros

  VALUE = next non-zero coefficient

- **Decoding Pipeline**
  To reconstruct the image:

  Read (RUNLENGTH, VALUE) pairs

  Expand back to a 64-element zigzag vector

  Convert using inverse zigzag to 8×8 matrix

  Dequantize:

$$\tilde{C}(i,j) = \hat{C}(i,j) \times Q(i,j)$$

  Apply 2D-IDCT to obtain spatial-domain block

  Combine all blocks back to the final image

III. Results

- **Reconstruction Quality**
  Both quantization tables successfully reconstruct recognizable images.

  Q1 preserves more detail but produces a larger encoded bitstream.

Q2 introduces stronger quantization artifacts (blocking and ringing) but yields higher compression.

- **Encoded Image Size Comparison**

Quantization Table	Encoded Size (Bytes)	Observations
Q1	51603	Higher quality, fewer zeros → longer RLE
Q2	19667	Heavier quantization → many zeros → much smaller size

IV. Discussion

- Using a zigzag scan significantly increases RLE efficiency by grouping high-frequency zeros together.

- The more aggressive quantization table (Q2) creates longer zero runs, which dramatically reduces encoded size.
- However, stronger quantization also reduces image quality due to higher information loss.
- This assignment demonstrates the core idea behind JPEG:
  Transform → Quantize → Entropy Encode,
  where quantization is the key factor balancing quality and compression.

V. Conclusion

- This assignment implemented block-based DCT, quantization, zigzag scanning, run-length encoding, and decoding to demonstrate the fundamental components of JPEG-style image compression. Experimental results confirmed the trade-off between bit-rate and image quality, showing how quantization strength influences RLE efficiency and reconstruction accuracy.
- The provided pipeline successfully encoded and decoded *lena.png* using two quantization tables, and the comparison highlights how transform-domain sparsity directly translates to compression gain.