

11. Given a dataset with repetitive patterns, design a simple Physical model for lossless compression. Justify how the model reduces data size and provide an example to demonstrate your approach.

ans: 1. Physical model: Run length encoding (RLE)

Concept

Instead of storing repeated data values individually, RLE stores:

- The value
- The number of times it repeats (run length)

Model representation

Each sequence is encoded as:

(value, count)

2. Compression Algorithm

encoding procedure

- Read the dataset sequentially
- Count consecutive occurrences of the same value
- Replace repeated values with a pair (value, count)
- Continue until the end of the dataset

Decoding procedure

- Read each (value, count) pair.
- Reconstruct the original data by repeating the value count times.

3. Justification of size reduction

let:

- original size = number of elements N
- compressed size = number of runs R

If data contains many repetitions:

$$R \ll N$$

So,

$$\text{compression Ratio} = N / (2R)$$

Because each run stores only:

- one value
- one count

Thus, repeated sequences are replaced by shorter representations, reducing storage.

Key insight

- works best when data has long repeated sequences.
- Eliminates redundancy by grouping identical values.

4. Example

original dataset

A A A A B B C C C C C A A

Step 1: Identify runs

- A repeated 4 times
- B repeated 2 times
- C repeated 5 times

- A repeated 2 times

Step 2: Encode

(A, 4), (B, 2), (C, 5), (A, 2)

Step 3: Size Comparison

- original length = 13 elements
- compressed length = 4 pairs = 8 values

Compression achieved

5. Code Snippet

```
def rle_encode(data):
```

```
    encoded = []
```

```
    count = 1
```

```
    for i in range(1, len(data)):
```

```
        if data[i] == data[i-1]:
```

```
            count += 1
```

```
        else:
```

```
            encoded.append((data[i-1], count))
```

```
            count = 1
```

```
    encoded.append((data[-1], count))
```

```
    return encoded
```

```
def rle_decode(encoded):
```

```
    decoded = []
```

```
    for value, count in encoded:
```

```
        decoded.extend([value] * count)
```

```
    return decoded
```


6. Advantages

- Simple to implement
- Lossless (original data fully recoverable)
- Effective for repetitive datasets.

7. Limitations

- Not efficient for non-repetitive data
- May increase size if data has no repetition.

8. Conclusion

The Run length encoding model compresses data by replacing repeated sequences with (value, count) pairs. It reduces data size by eliminating redundancy, making it highly effective for datasets with repetitive patterns while ensuring lossless reconstruction.