

Run-Length Encoding (RLE)

- BLM101 – Bilgisayar Mühendisliğine Giriş
- Dönem Projesi
- Ad Soyad: Ersin Polat
- Numara: 23360859701
- Bursa Teknik Üniversitesi

Sunum İçeriği

- • Veri nedir?
- • Bit düzeyinde temsil
- • Veri sıkıştırma
- • RLE algoritması
- • Python uygulaması

Veri Nedir?

- • Bilgisayarda saklanan bilgiler
- • Metin, resim ve ses verileri
- • Bitlerle temsil edilir

Bit Kavramı

- • Binary Digit
- • 0 ve 1 değerleri
- • En küçük veri birimi

Metin Verisinin Temsili

- • ASCII ve Unicode
- • Her karakter sayısaldır

Resim Verisinin Temsili

- • Piksel tabanlı yapı
- • Renkler bitlerle saklanır

Ses Verisinin Temsili

- • Analog → dijital
- • Örneklemeye

Veri Neden Sıkıştırılır?

- • Alan tasarrufu
- • Hızlı iletim

Veri Sıkıştırma Türleri

- • Kayıpsız
- • Kayıplı

Kayıpsız Sıkıştırma

- • Veri kaybı yok
- • RLE örnektir

Kayıplı Sıkıştırma

- • Veri kaybı vardır
- • MP3, JPEG

RLE Nedir?

- • Basit sıkıştırma
- • Tekrarları sayar

RLE Mantığı

- • Sayı + karakter
- • Ardışık tekrarlar

RLE Örnek

- AAAAABBBBCCDAA → 5A3B2C1D2A

RLE Avantajları

- • Basit
- • Hızlı

RLE Dezavantajları

- • Her veri için uygun değil

Python Projesi

- • Encode
- • Decode

Encode Fonksiyonu

- • Veriyi sıkıştırır

Decode Fonksiyonu

- • Orijinal veriyi döndürür

Sıkıştırma Oranı

- • Boyut karşılaştırması

Kod Yapısı

- • Fonksiyonlar
- • Yorum satırları

Kod Örneği

- `encode()`
- `decode()`

Program Çıktısı

- • Girdi
- • Çıktı

Sonuç

- • RLE basit ve etkilidir

Kaynaklar

- • Ders kitabı
- • Python Docs