

VERİ DEPOLAMA VE SIKIŞTIRMA ALGORİTMALARI



Bilgisayar Mühendisliği

Efe Yaman

24360859026

2.Grup

GitHub Linki:

[https://github.com/efeyaamn/BLM101 24360859026 EfeYaman](https://github.com/efeyaamn/BLM101_24360859026_EfeYaman)

GİRİŞ



**Veri nedir, neden önemli,
neden bu konu seçildi?**

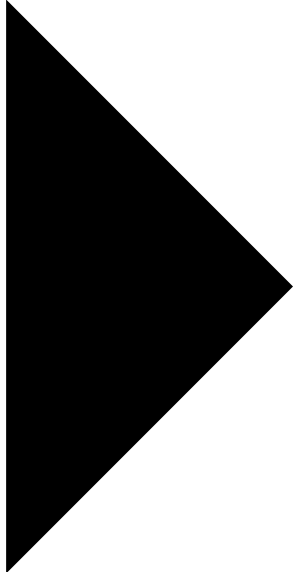
- Günümüzde veri miktarı artıyor**
- Depolama ve iletim problemi var**
- Veri sıkıştırma bu yüzden önemli**



Veri nedir, neden önemli, neden bu konu seçildi?

- Günümüzde veri miktarı artıyor**
- Depolama ve iletim problemi var**
- Veri sıkıştırma bu yüzden önemli**

Veri Nedir?



Veri, bilgisayarların işleyebildiği ham bilgidir. Metinler, sayılar, resimler ve sesler veri örnekleridir. Bilgisayarlar tüm verileri ikili (0 ve 1) sistemde saklar.



Bit ve Byte Kavramı

Bit, bilgisayardaki en küçük veri birimidir ve 0 veya 1 değerini alır. 8 bit bir araya gelerek 1 byte oluşturur. Tüm veriler byte'lar üzerinden depolanır.



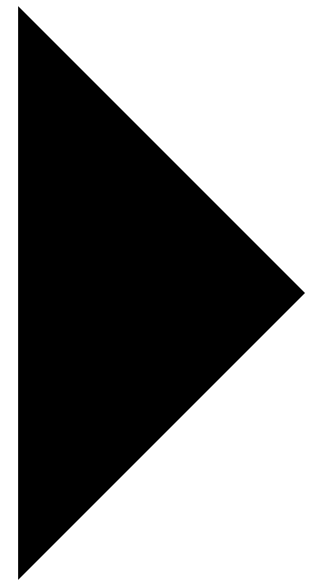
Bilgisayarlarda Veri Temsili

Bilgisayarlar metin, resim ve ses gibi farklı veri türlerini bit dizileri halinde saklar. Bu temsiller standartlara (ASCII, Unicode vb.) göre yapılır.



Metin Verilerinin Temsili

Metin verileri karakter kodlama sistemleri ile temsil edilir. Örneğin ASCII sisteminde her harf 8 bit ile gösterilir.



Resim Verilerinin Temsili

Resimler piksellerden oluşur. Her pikselin bir renk değeri vardır ve bu değerler bitler ile ifade edilir.



Ses Verilerinin Temsili

Ses verileri, ses dalgalarının belirli aralıklarla örneklenmesiyle dijital hale getirilir. Bu örnekler bit dizileri olarak saklanır.



Veri Sıkıştırma Nedir

Veri sıkıştırma, bir verinin kapladığı alanı azaltmak için kullanılan yöntemler bütünüdür. Amaç, depolama ve iletim maliyetlerini düşürmektir.



Veri Sıkıştırılmaya Neden İhtiyaç Duyulur?

Daha az depolama alanı kullanmak, internet üzerinden daha hızlı veri transferi yapmak ve maliyetleri azaltmak için veri sıkıştırma gereklidir.



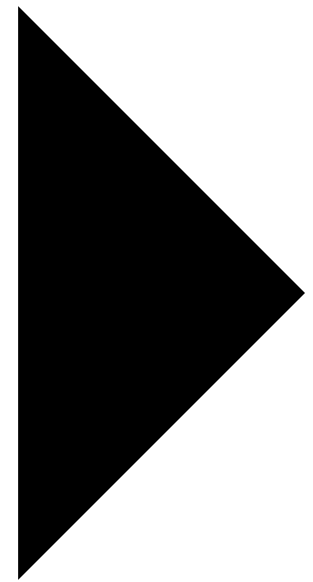
Kayıplı ve Kayıpsız Sıkıştırma

Kayıplı sıkıştırmada veri geri açıldığında birebir aynı olmaz. Kayıpsız sıkıştırmada ise veri tamamen eski haline döner. RLE kayıpsızdır.



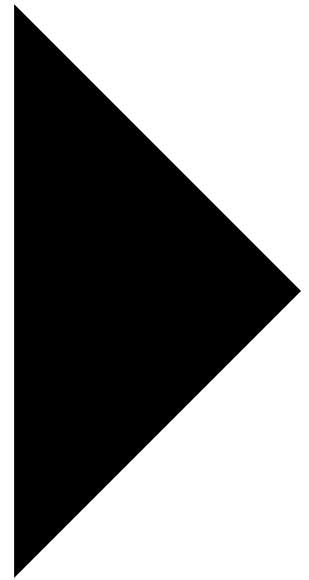
RLE (Run-Length Encoding) Nedir?

RLE, ardışık tekrar eden karakterleri sayılarıyla birlikte saklayan basit bir sıkıştırma algoritmasıdır.



RLE Algoritmasının Çalışma Mantığı

**Arka arkaya gelen aynı karakterler sayılır ve sayı +
karakter şeklinde yazılır.**



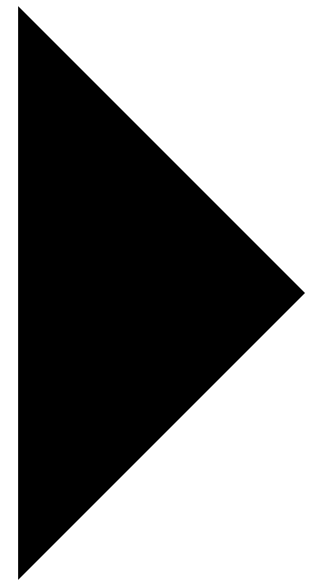
RLE Encode Örneği

Örnek giriş: AAABBBCCDAA Sıkıştırılmış hali: 3A3B2C1D2A



RLE Decode Örneği

Decode işlemi, sıkıştırılmış veriyi tekrar eski haline döndürme işlemidir.



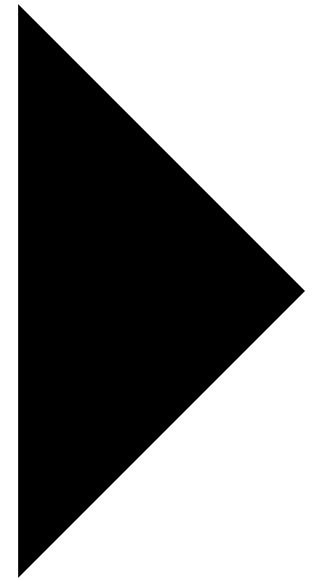
RLE Algoritmasının Avantajları

Basit, hızlı ve kolay uygulanabilir bir algoritmadır.



RLE Algoritmasının Dezavantajları

Tekrar eden veriler azsa sıkıştırma yerine veri boyutu artabilir.



Python ile RLE Uygulaması

Bu projede RLE algoritması Python programlama dili kullanılarak uygulanmıştır.



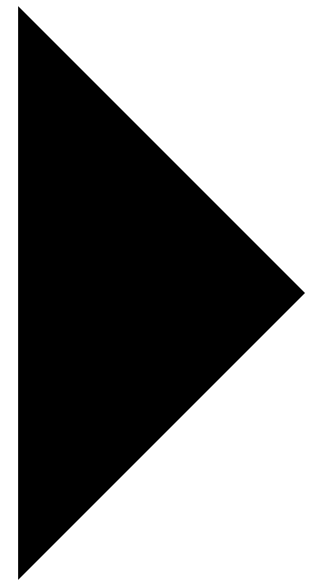
Encode Fonksiyonunun Görevi

Kullanıcıdan alınan metni RLE algoritması ile sıkıştırır.



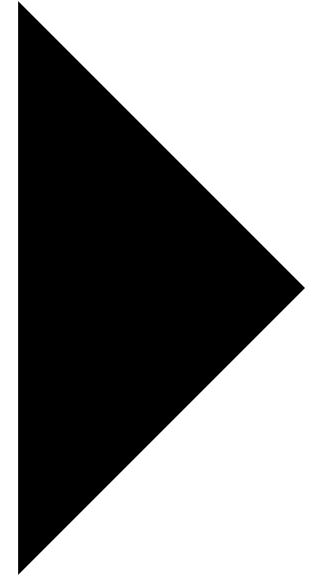
Decode Fonksiyonunun Görevi

**Sıkıştırılmış metni çözerek orijinal haline
geri döndürür.**



Sıkıştırma Oranı Nedir?

Sıkıştırma oranı, sıkıştırılmış veri boyutunun orijinal veri boyutuna oranıdır.



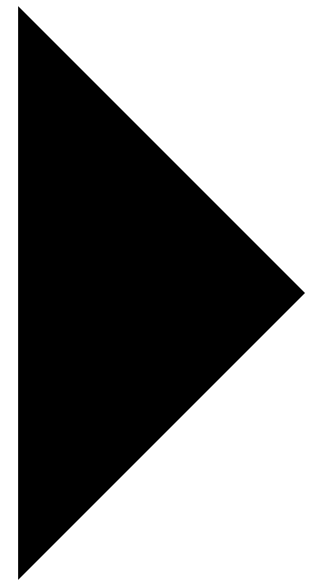
Sıkıştırma Oranı Hesaplama

$(\text{Orijinal Uzunluk} - \text{Sıkıştırılmış Uzunluk}) / \text{Orijinal Uzunluk} * 100$



Programın Çalışma Akışı

**Kullanıcıdan veri alınır, encode edilir, decode edilir
ve sıkıştırma oranı hesaplanır.**



Kod İçerisinde Kullanılan Yapılar

**Fonksiyonlar, döngüler ve koşul yapıları
kullanılmıştır.**



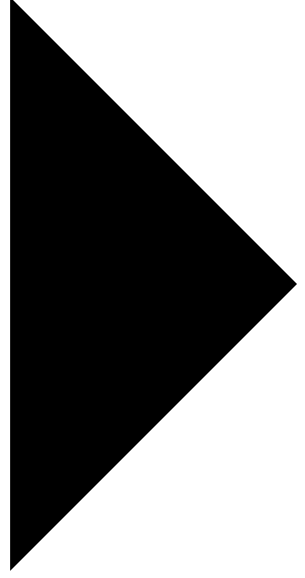
Proje Sonuçları

RLE algoritmasının basit veri kümelerinde etkili olduğu görülmüştür.



Genel Değerlendirme

Bu proje sayesinde veri sıkıştırma mantığı ve RLE algoritması öğrenilmiştir.



Kaynakça

Ders kitabı ve ders notları kullanılmıştır.