

BÖLÜM 1: BİLGİNİN TEMSİLİ

BİLGİSAYAR BİLİMLERİNE GİRİŞ: VERİ DEPOLAMA VE SIKIŞTIRMA TEKNOLOJİLERİ

**Brookshear & Brylow (Bölüm 1.4 - 1.9) Işığında Bilgi Temsili ve RLE
Algoritması Uygulaması**

**Bu sunumda, dijital verinin temellerinden uygulama aşamasına kadar olan
süreçten bahsedeceğiz.**



Ad Soyad: Meryem Sümeyye Aysal
Öğrenci No: 25360859321

HER ŞEY NASIL 1 VE 0 OLDU?

- Bilgisayarların temel yapı taşı olan transistörler, sadece "açık" (1) veya "kapalı" (0) durumlarını anlayabilir.
- Bu en küçük bilgi birimine Bit (Binary Digit) adı verilir.
- Metinler, görüntüler ve sesler bu iki durumun karmaşık kombinasyonları kullanılarak dijitalleştirilir.
- Veri temsili, fiziksel bir olgunun (ışık, ses, elektrik) matematiksel bir modele dönüştürülmesidir.

Bunları biliyor
muydunuz?

Uzaydan Gelen Veri
İçerik: "Voyager 1 uzay aracı, Dünya'dan milyarlarca kilometre uzakta olmasına rağmen çektiği fotoğrafları bize hala **0 ve 1** dizileri (Binary) olarak gönderiyor. Veri sıkıştırma algoritmaları (sunumumuzun ilerleyen kısımlarında göreceğimiz üzere), bu verilerin sınırlı bant genişliği ile Dünya'ya ulaşmasını sağlayan gizli kahramanlardır."



METİNLERİN TEMSİLİ: ASCII STANDARTLARI

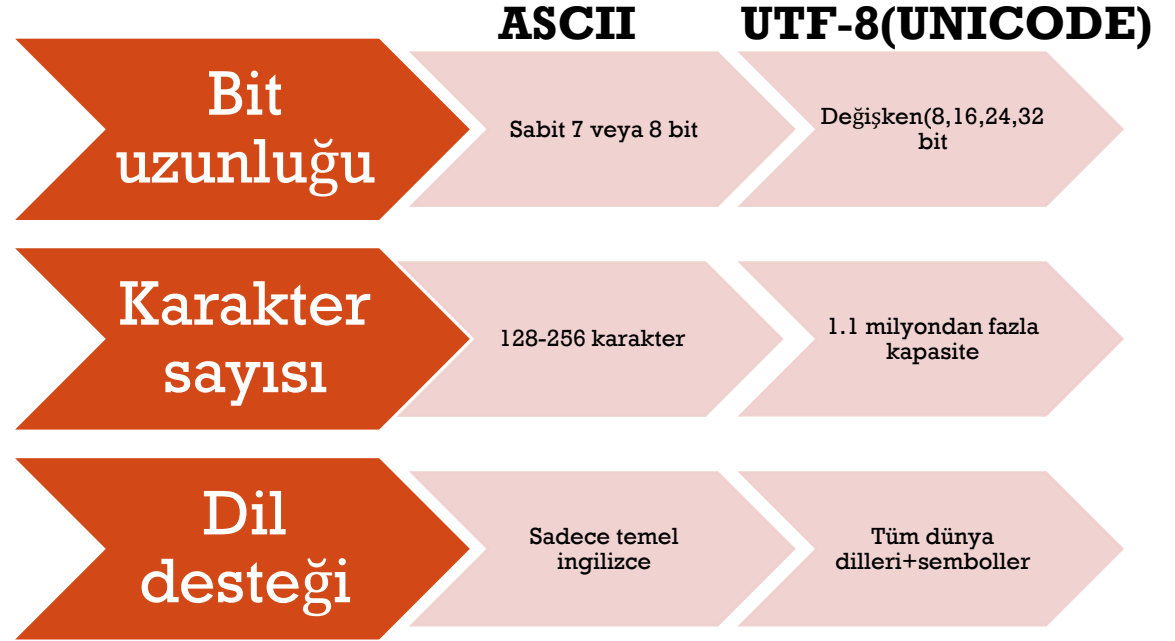
- American Standard Code for Information Interchange (ASCII), bilgisayarlar arası metin alışverişi için oluşturulmuş ilk yaygın standarttır.
- Mekanizma: Her karakter (büyük-küçük harf, rakam, noktalama) 7 veya 8 bitlik bir sayı dizisiyle eşleşir.
- Örnek: 'A' harfi 65 (01000001), 'a' harfi ise 97 (01100001) olarak kodlanır.
- Kısıtlama: Sadece İngiliz alfabesi ve temel kontrol karakterlerini kapsadığı için küresel ihtiyaçlarda yetersiz kalmıştır.



MODERN DÜNYA STANDARTI: UNICODE VE UTF-8

- **Evrensel Metin Temsili: Unicode**

- Kapsam: Dünyadaki tüm dilleri, antik alfabeleri, matematiksel sembolleri ve emojileri tek bir çatı altında toplar.
- UTF-8: Unicode karakterlerini kodlamak için en popüler yöntemdir. Verinin boyutuna göre değişken uzunlukta (1 ila 4 byte) kodlama yapar.
- Uyumluluk: İlk 128 karakteri ASCII ile tamamen aynıdır, bu da eski sistemlerle mükemmel uyum sağlar.
- Önem: Modern internetin ve yazılımların metin tabanlı omurgasını oluşturur.



SAYISAL VERİLERİN SAKLANMASI - TAM SAYILAR

- BELLEKTE TAMSAYILAR(INTEGERS)
- İkilik Taban: Sayılar onluk sistemden (decimal) ikilik sisteme (binary) çevrilerek saklanır.
- İşaretli Sayılar (Two's Complement): Negatif sayıları temsil etmek için en yaygın yöntemdir. En soldaki bit (MSB) işaret biti olarak işlev görür.
- Sınırlar: 8-bit bir sistemde -128 ile +127 arasındaki sayılar saklanabilirken, 32-bit sistemlerde milyarlarca sayı temsil edilebilir.
- Taşma (Overflow): Belirlenen bit kapasitesinin üzerine çıkıldığında yanlış sonuçlar doğuran bir hata türüdür.



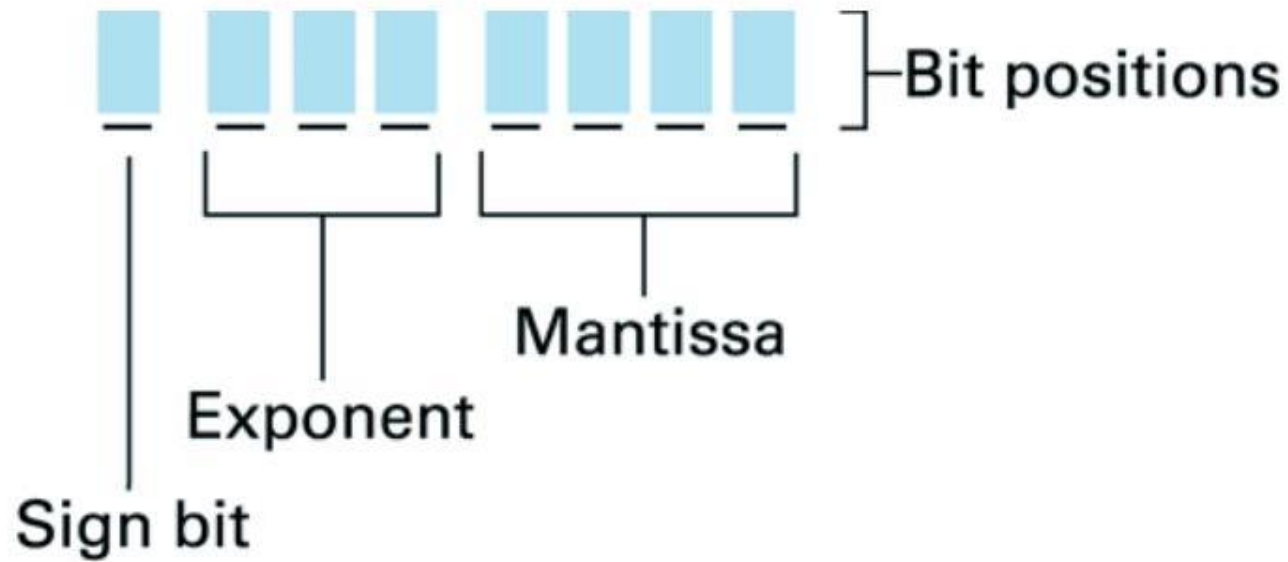
VISUALIZING MEMORY: INTEGERS VS. FLOATING POINT



"Data representation is the foundation of computing."



Kayan Nokta Gösterim Bileşenleri



HASSAS HESAPLAMALAR: ONDALIKLI SAYILAR

Kayan Noktalı Sayılar (Floating-Point)

Standart: Bilgisayarlar ondalıklı sayıları IEEE 754 standardına göre saklar.

Bileşenler: Sayı üç parçaya bölünür: İşaret (Sign), Üs (Exponent) ve Mantis/Kesir (Mantissa).

Çalışma Mantığı: Bilimsel gösterime (2.99×10^8) benzer şekilde, çok büyük veya çok küçük sayıların dar bir alanda saklanmasına olanak tanır.

Yuvarlama Hataları: Bazı ondalıklı sayılar (örneğin 0.1) ikilik sistemde sonsuz bir dizi oluşturduğu için ufak hassasiyet kayıpları yaşanabilir.



GÖRSEL VERİNİN TEMSİLİ - PİKSEL VE BIT-MAP

- **Görüntülerin Dijital Temsili: Bit-Map Teknolojisi**
- **Piksel:** Bir görüntüyü oluşturan en küçük renk hücresidir (Picture Element).
- **Renk Derinliği:** Her pikselin rengini belirlemek için kullanılan bit sayısıdır.
- **RGB Modeli:** Kırmızı (Red), Yeşil (Green) ve Mavi (Blue) renklerinin farklı yoğunluklarda birleşimiyle milyonlarca renk üretilir.
- **Örnek:** 24-bit renk sisteminde her ana renk için 8 bit ayrılır (2^{24} = 16.7 milyon renk).



VECTOR GRAPHICS: MATHEMATICAL PRECISION



SCALABILITY: NO PIXELATION



$$\bigcirc = \frac{4}{2} - 2 = \frac{1}{x - 1}$$

DEFINITION: GEOMETRIC
SHAPES + FORMULAS

VECTOR FILES (SVG, AI, EPS) ARE INDEPENDENT OF RESOLUTION.

GEOMETRİK TEMSİL: VEKTÖR GRAFİKLERİ

Tanım: Görüntü, noktalar ve renkler yerine matematiksel formüllerle tanımlanır.

Mekanizma: "X noktasından Y noktasına bir eğri çiz ve içini maviye boya" gibi komutlar saklanır.

Ölçeklenebilirlik: Görüntü ne kadar büyütülürse büyütülsün matematiksel olarak yeniden hesaplandığı için pikselleşme (bozulma) olmaz.

Kullanım Alanı: Logolar, fontlar (yazı tipleri) ve teknik çizimler (CAD).



SESİN SAYISALLAŞTIRILMASI

- **Analogtan Dijitale: Ses Örneklemesi (Sampling)**
- **Sürekli Dalga:** Ses doğal haliyle sürekli bir dalgadır; bilgisayar ise kesikli veriyi anlar.
- **Örnekleme:** Ses dalgasının yüksekliği saniyede binlerce kez ölçülür (Örn: CD kalitesi için saniyede 44.100 kez).
- **Kuantizasyon:** Örneklenen değerlerin en yakın sayısal seviyeye yuvarlanması işlemidir.
- **Bit Oranı:** Saniyedeki örnek sayısı ve her örnek için kullanılan bit sayısı arttıkça kalite artar.





Low Sampling Rate
(Low Quality)

High Sampling Rate
(CD Quality)



SIRA SENDE!

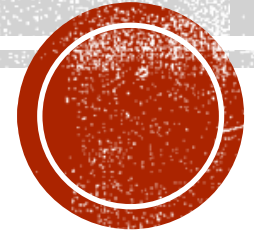
"Gördüğünüz gibi, sesi ne kadar kaliteli kaydederseniz o kadar çok veri üretiriz. Peki, bu devasa verileri internette nasıl hızlıca gönderiyoruz?"

Conversion to Bits: The Foundation of Audio Storage



BÖLÜM 2: VERİ SIKIŞTIRMA

(BÖLÜM 1.9)



VERİ SIKIŞTIRMAYA GİRİŞ

- **Neden Veriyi Küçültmeliyiz?**
- **Tanım:** Verinin kapladığı bit sayısını, içindeki bilgiyi koruyarak azaltma işlemidir.
- **Bant Genişliği:** İnternet üzerinden video izlerken veya dosya gönderirken zaman kazanmamızı sağlar.
- **Depolama Maliyeti:** Sabit disklerde veya bulut sunucularında daha fazla dosya saklamaya imkan tanır.
- **Enerji Verimliliği:** Daha az verinin işlenmesi ve transfer edilmesi, mobil cihazlarda pil tasarrufu sağlar.



KAYIPLI VE KAYIPSIZ SIKIŞTIRMA

KAYIPLI SIKIŞTIRMA

- İlke: Sıkıştırılan veri açıldığında (decompressed), orijinal verinin bit bazında aynısı elde edilir.
- Kullanım: Metin belgeleri, kaynak kodlar ve veri tabanları gibi tek bir karakterin bile değişmemesi gereken durumlarda kullanılır.
- Teknik: Verideki istatistiksel fazlalıkları (redundancy) ortadan kaldırır.
- Formatlar: .ZIP, .RAR, .PNG, .FLAC.

KAYIPSIZ SIKIŞTIRMA

- İlke: İnsan duyularının (göz ve kulak) fark edemeyeceği detaylar kalıcı olarak silinir.
- Verimlilik: Kayıpsız yöntemlere göre çok daha yüksek (10-20 kat) sıkıştırma oranları sunar.
- Kullanım: Fotoğraf, müzik ve video akış servisleri (YouTube, Netflix, Spotify).
- Formatlar: .JPEG (görüntü), .MP3 (ses), .MP4 (video).



Lossless Compression



Perfect Quality, Larger Files



Verifiable Security

VS.

Lossy Compression



Reduced Quality, Tiny Files



Deletion Transfer

Size Comparison



The fundamental trade-off: fidelity vs. space.

SIKIŞTIRMA ORANI VE BAŞARI KRİTERLERİ

■ *Verimliliğin Ölçülmesi;*

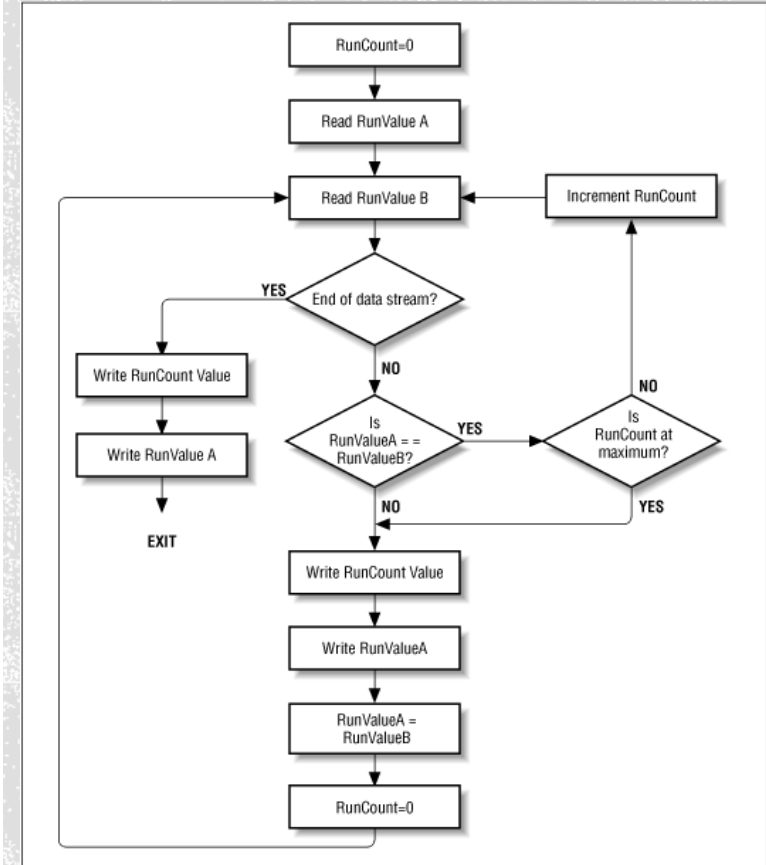
FORMUL:SIKIŞTIRMA ORANI= $\frac{\text{SIKIŞTIRILMIŞ BOYUT}}{\text{ORJİNAL BOYUT}}$.100

- **Yüzde:** Bir veriyi %40 sıkıştırmak, boyutun orijinalin %60'ına inmesi demektir.
- **Kalite Kaybı:** Kayıplı sıkıştırmada, "Sinyal-Gürültü Oranı" (SNR) gibi metriklerle kalite kaybı ölçülür.
- **Hız Faktörü:** Sıkıştırma algoritmasının ne kadar hızlı çalıştığı (real-time sistemler için kritik) önemli bir başarı kriteridir.



BOLUM 3: RUN-LENGTH ENCODING (RLE) DERİNLEMESİ

- RLE Algoritması: Temel Mantık
- Mantık: Veri içinde art arda gelen aynı sembolleri tek tek saklamak yerine "kaç tane olduğu + sembolün kendisi" şeklinde saklar.
- En Basit Örnek: BBBBBB (6 byte) yerine 6B (2 byte) yazılır. İşlem Basamakları: 1. Veri dizisini oku, 2. Mevcut karakteri ve sayısını belirle, 3. Karakter değiştiğinde eski miktar ve karakteri kaydet.
- Uygulama: Faks makineleri ve basit grafik formatlarında (BMP'nin bazı türleri) yaygın kullanılır.



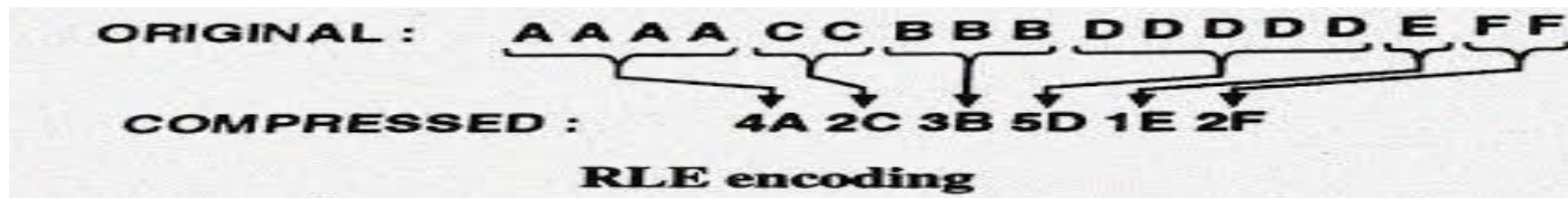
RLE'NİN PERFORMANS ANALİZİ

- ***Ne Zaman RLE Kullanılmalı?***
- En İyi Durum: Çok fazla ardışık tekrarın olduğu veriler (Örn: Geniş beyaz alanları olan bir resim dosyası).
- En Kötü Durum: Hiç tekrarın olmadığı veriler (Örn: ABCDEFG). Bu durumda çıktı 1A1B1C1D1E1F1G olur ve dosya boyutu iki katına çıkar.
- Karar Mekanizması: Modern sistemler, veriyi analiz ederek eğer RLE boyutu büyütecekse sıkıştırmadan olduğu gibi saklar.



BÖLÜM 4: PYTHON İLE RLE UYGULAMASI

- *Algoritma Tasarımı ve İşleyiş Mantığı*
- RLE Sıkıştırma Algoritmasının Adımları:
- Girdi Analizi: Program, kullanıcıdan aldığı metni karakter karakter tarar. Ardışık Sayım: Bir karakterin arka arkaya kaç kez tekrarlandığını belirlemek için bir sayaç kullanılır.
- Kod Oluşturma: Karakter değiştiğinde, o ana kadar olan sayaç + karakter bilgisi sonuç dizisine eklenir.
- Veri Tipi Dönüşümü: Sayısal değerler string formatına çevrilerek karakterin yanına eklenir.



- `def rle_encode(data):`
- `if not data: return ""`
- `encoding = ""`
- `prev_char = data[0]`
- `count = 1`
- `for i in range(1, len(data)):`
- `if data[i] == prev_char:`
- `count += 1`
- `else:`
- `encoding += str(count) + prev_char`
- `prev_char = data[i]`
- `count = 1`
- `encoding += str(count) + prev_char`
- `return encoding`

PYTHON KOD BLOĞU - SIKIŞTIRMA (ENCODE)

Kayıpsız Sıkıştırma Fonksiyonu:

Ekranda örnek bir kod
bulunmaktadır. Beraber
inceleyelim!



Sıkıştırılmış Veri
(Compressed)

5A 3B 2C

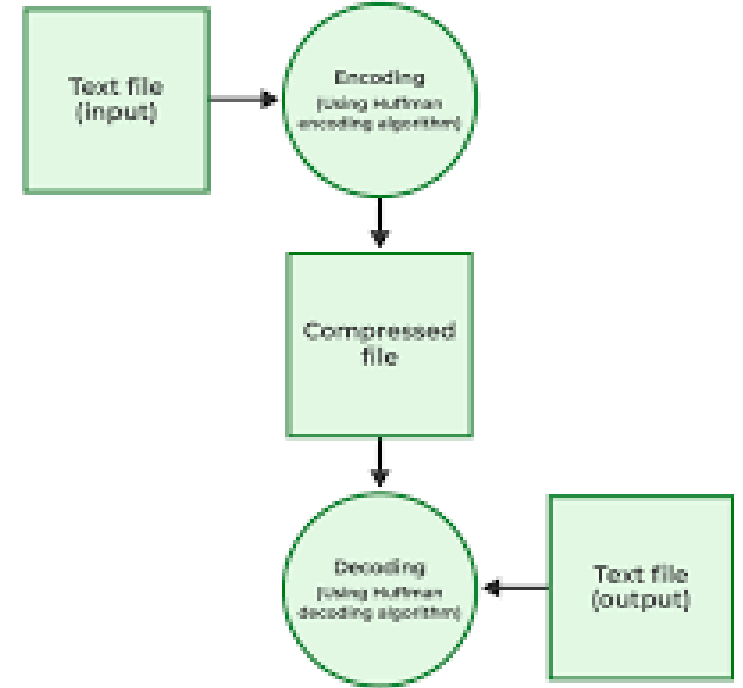
DECODING
(Geri Açma)

Orjinal Veri
(Decompressed)

AAAAABBBCC

- Sıkıştırılmış Veriyi Orijinaline Dönüştürme
- Sıkıştırma kadar önemli olan adım, veriyi hatasız geri açmaktır.
- Program, sayıları okur ve yanındaki karakteri o sayı kadar tekrar yazdırır.
- Önemli: Eğer 5A okunuyorsa, çıktı AAAAA olur. Bu, verinin Kayıpsız (Lossless) olduğunu kanıtlar.

PYTHON KOD BLOĞU - GERİ AÇMA (DECODE):

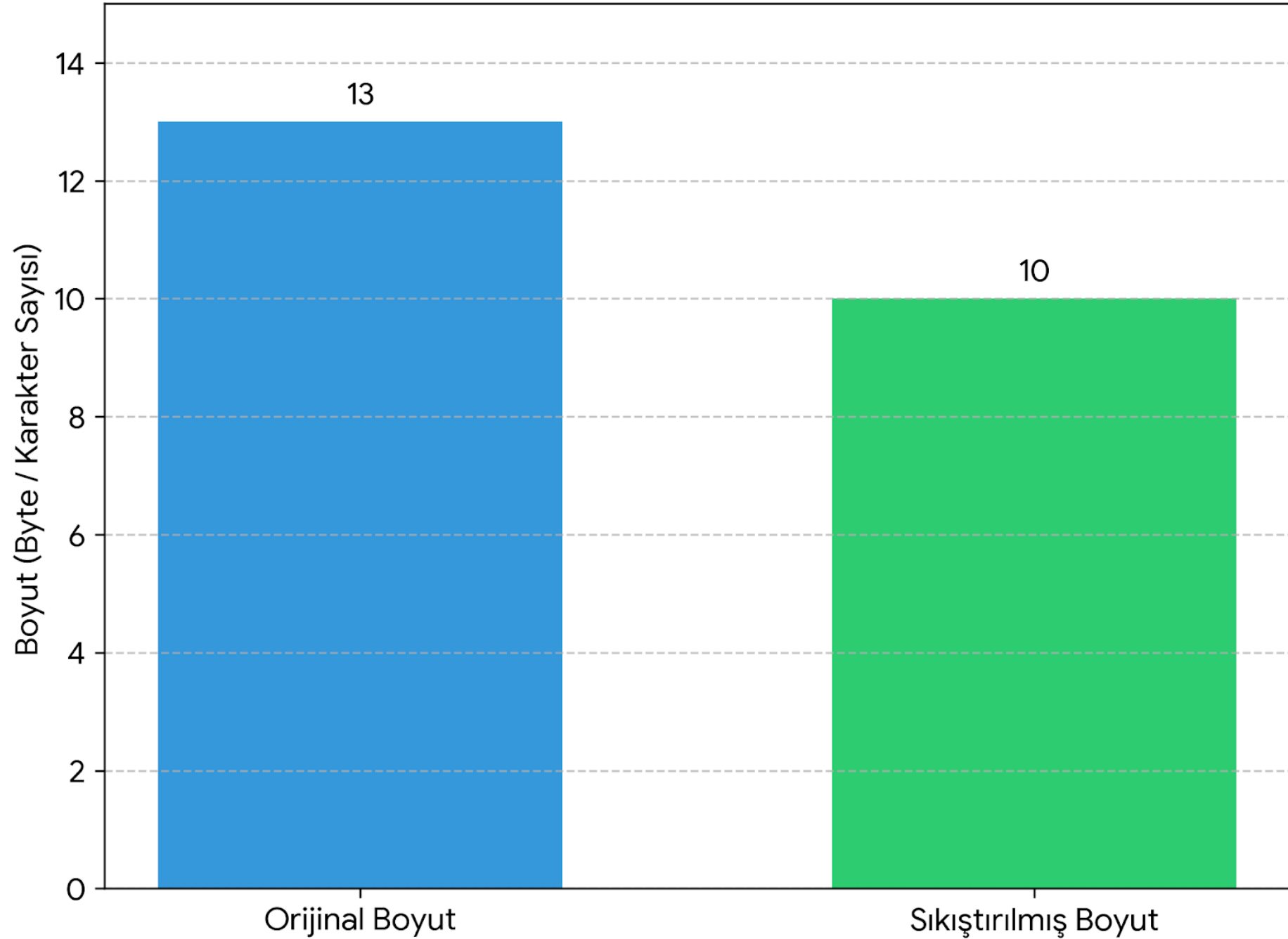


BÖLÜM 5: SONUÇ VE ANALİZ

- ***Performans Analizi: Ne Kadar Tasarruf Ettik?***
- Sıkıştırma Verimliliğinin Ölçülmesi:
- Örnek Senaryo: "AAAAABBBCCDAA" (13 Karakter / 13 Byte)
- RLE Sonucu: "5A3B2C1D2A" (10 Karakter / 10 Byte)
- Hesaplama: $\text{Oran} = (10/13) \times 100 = \%76.9$
- Yorum: Veri boyutu yaklaşık %23 oranında küçültülmüştür.
- Diğer sayfadaki sütun grafiğinde daha kolay bir şekilde incelenebilir.



RLE Sıkıştırma Performans Analizi (AAAAABBBBCCDAA)



RLE'NİN SINIRLARI VE RİSKLER

- ***Her Veri Sıkıştırılabilir mi?***
- Negatif Verimlilik: Eğer veri çok değişkense (Örn: "ABCDE"), RLE sonucu "1A1B1C1D1E" olur.
- Boyut Artışı: 5 karakterlik veri 10 karaktere çıkar. Bu, RLE'nin en büyük dezavantajıdır.
- Çözüm: Modern yazılımlar, sıkıştırma sonrası boyut büyüyorsa orijinal dosyayı tutmayı tercih eder.



GENEL DEĞERLENDİRME VE GELECEK

- Veri Temsilinden Sıkıştırmaya: Neler Öğrendik?
- Bilgisayarın temelinde yatan 0 ve 1'lerin (Bölüm 1.4) sadece sayıları değil, karmaşık algoritmaları da nasıl şekillendirdiğini gördük.
- Veri sıkıştırma (Bölüm 1.9), günümüzün "Big Data" çağında depolama ve bant genişliği sorunlarını çözen kritik bir araçtır.
- RLE basit bir başlangıçtır; ancak JPEG ve MP3 gibi modern yöntemlerin temel mantığını anlamamızı sağlar.



DİNLEDİĞİNİZ İÇİN TEŞEKKÜRLER!

Bursa Teknik Üniversitesi Bilgisayar Mühendisliği
Ad Soyad : Meryem Sümeyye Aysal
Öğrenci Numarası:25360859321

