BURSA TEKNİK ÜNİVERSİTESİ
Doğa Bilimleri Mimarlık ve Mühendislik Fakültesi
Bilgisayar Mühendisliği

VERİ DEPOLAMA VE SIKIŞTIRMA ALGORİTMALARI

Bilgisayar Bilimlerine Giriş Dersi Dönem Projesi

Hazırlayan: Gülşen Çeken
Öğrenci No: 25360859006
Danışman/Hoca: Prof. Dr. TURGAY TUGAY BİLGİN

BURSA, 2026

Bölüm 1 : Veri depolama

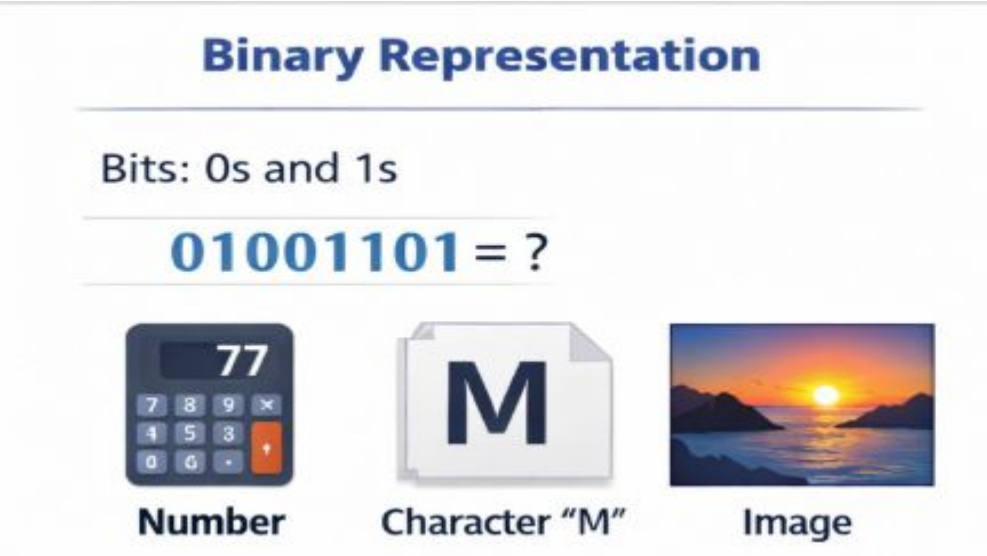
- 1.4 Bit Desenleri Olarak Bilgi Gösterimi
- 1.5 İkilik Sistem
- 1.6 Tam Sayıları Depolama
- 1.7 Kesirleri Depolama
- 1.8 Veri ve Programlama
- 1.9 Veri Sıkıştırma

1.4 Bit Desenleri Olarak Bilgi Gösterimi

Bit, 0 veya 1 değerini alabilen en küçük veri birimidir. Bu bitlerin belirli bir sırada dizilmesiyle oluşan bit desenleri; sayılar, karakterler, görüntüler ve sesler gibi farklı veri türlerini temsil edebilir.

Ancak bir bit deseninin anlamı, onu yorumlayan sistemin veri türüne bağlıdır. Aynı bit dizisi, farklı bağlamlarda farklı bilgileri ifade edebilir. Bu durum, veri türlerinin bilgisayar bilimlerindeki önemini ortaya koyar.

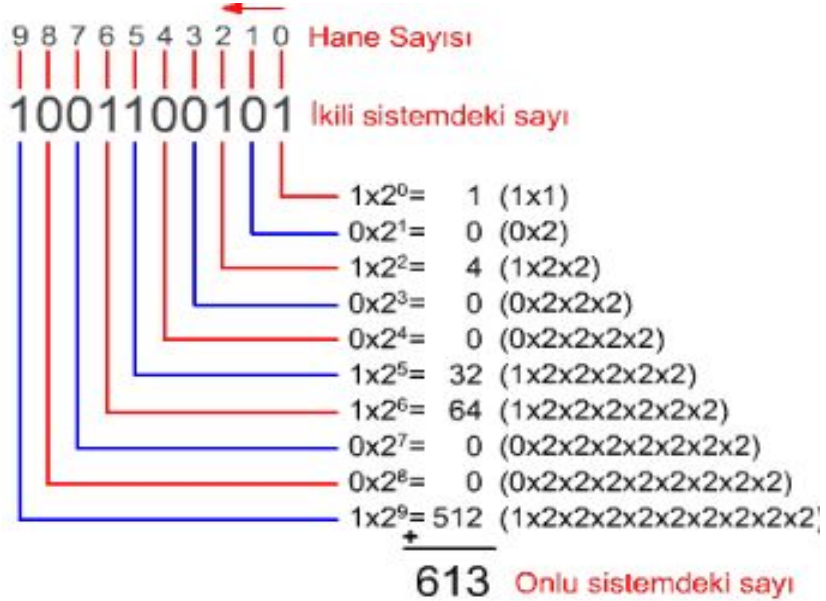
- **Ses:** Analog dalgalar örnekleme (sampling) yoluyla dijitalleştirilir.
- **Metin:** Karakterler ASCII veya Unicode ile kodlanır.
- **Görüntü:** Renkler RGB değerleri ile temsil edilir.

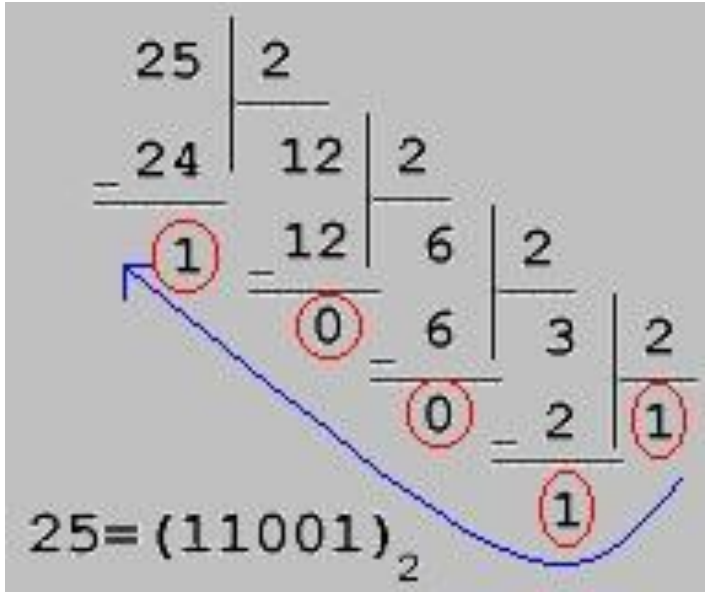


1.5 İkilik Sistem (Binary System)

İkilik sayılar 2'lik tabanı kullanır ve 1'ler ile 0'lar serisi olarak sunulur.

- **Hesaplama:** Bir sayının değeri, sağdan sola doğru taranarak hesaplanır.
- **Formül:** Her basamak, 2'nin o basamağın konumuna denk gelen kuvvetiyle çarpılır.

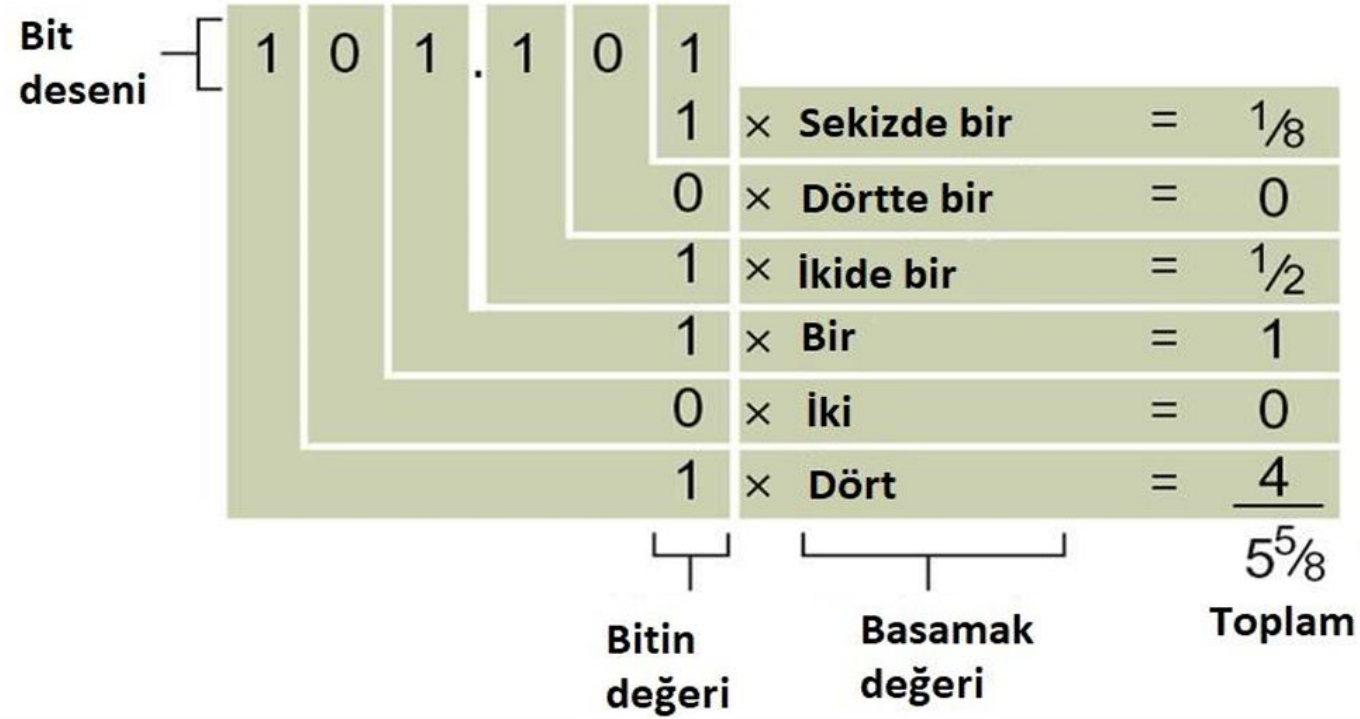

9 8 7 6 5 4 3 2 1 0 Hane Sayısı
1001100101 İkili sistemdeki sayı
1x2⁰= 1 (1x1)
0x2¹= 0 (0x2)
1x2²= 4 (1x2x2)
0x2³= 0 (0x2x2x2)
0x2⁴= 0 (0x2x2x2x2)
1x2⁵= 32 (1x2x2x2x2x2)
1x2⁶= 64 (1x2x2x2x2x2x2)
0x2⁷= 0 (0x2x2x2x2x2x2x2)
0x2⁸= 0 (0x2x2x2x2x2x2x2x2)
1x2⁹= 512 (1x2x2x2x2x2x2x2x2x2)
613 Onlu sistemdeki sayı


25 | 2
24 | 12 | 2
1 | 12 | 6 | 2
0 | 6 | 3 | 2
0 | 2 | 1
1
25=(11001)2

1.5.1 İkilik Sistemde Toplama

$$\begin{array}{r} 0 \\ +\,0 \\ \hline 0 \end{array} \qquad \begin{array}{r} 1 \\ +\,0 \\ \hline 1 \end{array} \qquad \begin{array}{r} 0 \\ +\,1 \\ \hline 1 \end{array} \qquad \begin{array}{r} 1 \\ +\,1 \\ \hline 10 \end{array}$$

1.5.2 İkilik Sistemde Kesirler



1.6 Tam Sayıların Gösterimi

- **İkinin Tümleyeni Gösterimi:** Tam sayıları göstermek için en popüler sistem türüdür.
- **Fazlalık (Excess) Gösterimi:** İkinin tümleyeni gösteriminde olduğu gibi fazlalık gösterim sistemindeki değerlerin her biri aynı uzunluktaki bir bit deseni tarafından gösterilir.

Her ikisinde de taşma hataları olabilir.

1.6.1 İkinin Tümleyeni Gösterimi

İki'nin tümleyeni, bilgisayarlarda **negatif tam sayıların** bit desenleriyle gösterilmesini sağlayan bir yöntemdir. En soldaki bit **işaret bitidir**; 0 pozitif, 1 negatif sayıyı ifade eder. Negatif bir sayı elde etmek için sayının ikilik gösteriminin bitleri ters çevrilir ve **1 eklenir**. Bu yöntem, çıkarma işlemlerinin toplama yoluyla yapılmasını sağlar.

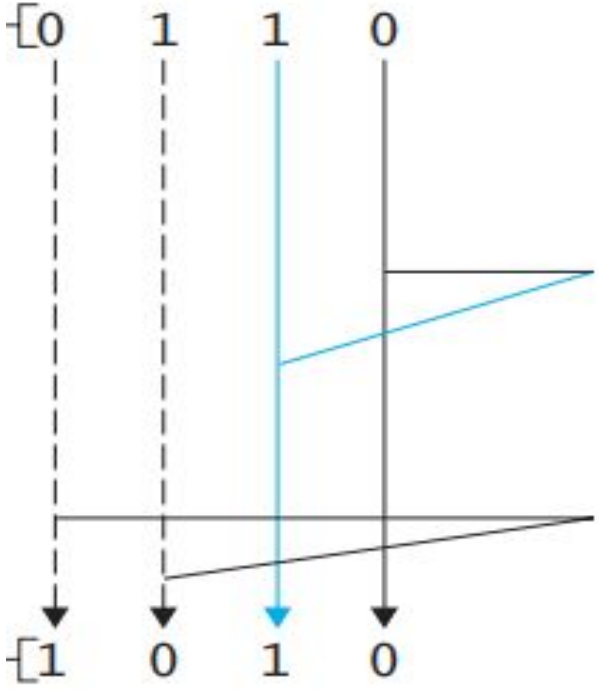
a.Üç bit uzunluğundaki desenlerde gösterim

Bit deseni	Gösterdiği değer
011	3
010	2
001	1
000	0
111	−1
110	−2
101	−3
100	−4

b.Dört bit uzunluğundaki desenlerde gösterim

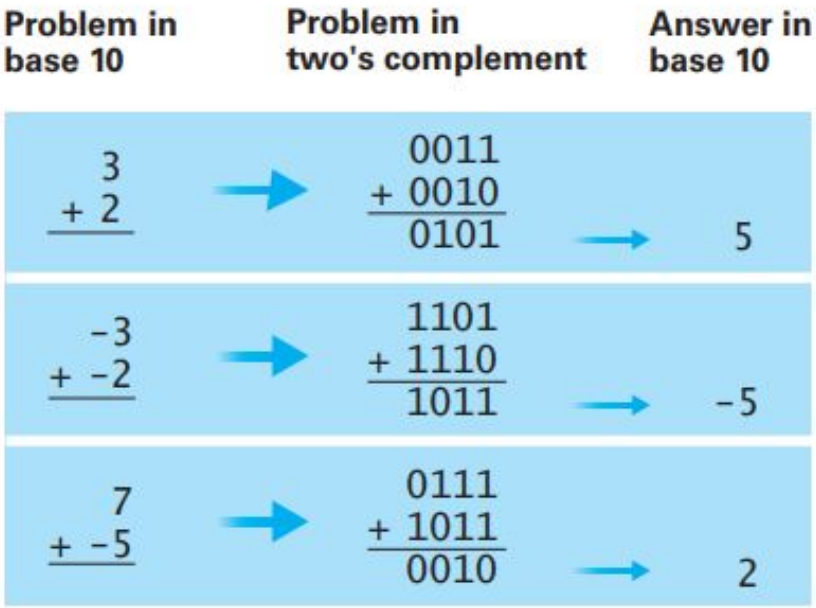
Bit deseni	Gösterdiği değer
0111	7
0110	6
0101	5
0100	4
0011	3
0010	2
0001	1
0000	0
1111	−1
1110	−2
1101	−3
1100	−4
1011	−5
1010	−6
1001	−7
1000	−8

Dört bit kullanarak 6 için ikinin tümleyeni gösterimi



Dört bit kullanarak -6 için ikinin tümleyeni gösterimi

1.6.1.1 İkinin Tümleyeni Gösteriminde Toplama



7 − 5 → 0111 − 0101 → 0111 + 1011 = 0010 → 2

Taşma(Overflow): Bir değer temsil edilemeyecek kadar büyük olduğunda oluşur.

Kesme(Truncation): Bir değer doğru şekilde temsil edilmediğinde ortaya çıkar.

1.6.1.2 Fazlalık (Excess) Gösterimi

Tam sayıları göstermenin bir başka yöntemi fazlalık gösterimidir.

Fazlalık gösteriminin ana fikri, sayı doğrusunu belirli bir miktar sağa kaydırmaktır. Negatif sayıları temsil etmek için, tüm sayılara sabit bir **"fazlalık" (bias)** değeri eklenir. Böylece temsil edilmek istenen en küçük negatif sayı bile pozitif bir değere dönüştürülür.

Neden Kullanılır?

Fazlalık gösteriminin en büyük avantajı **karşılaştırma kolaylığıdır**.

- İkilik desenler (0000'dan 1111'e kadar) küçükten büyüğe doğru sıralandığında, temsil ettikleri gerçek sayılar da (-8'den +7'ye doğru) tam olarak aynı sırayla büyür.
- Bu özellik, bilgisayarın iki sayıyı karşılaştırırken karmaşık işaret kontrolleri yapmadan standart "unsigned" (işaretsiz) karşılaştırma mantığını kullanabilmesini sağlar.

Özetle Farkı:

- **İkinin Tümleyeni:** Sayının işaretini değiştirmek için bitleri ters çevirip 1 ekleriz.
- **Fazlalık Gösterimi:** Sayıya sabit bir sayı ekleyerek her şeyi pozitif bir aralığa taşırız.

Formül: Depolanan Değer = Gerçek Sayı + Fazlalık (Bias)

2. Excess-8 (4-Bit) Örneği

4 bitlik bir sistemde toplam $2^4 = 16$ farklı kombinasyon vardır. Genellikle bu sistemde orta nokta olan **8**, fazlalık (bias) değeri olarak seçilir.

Gerçek Sayı (Ondalık)	Hesaplama (Sayı + 8)	Depolanan İkilik Desen
+7 (En büyük)	7 + 8 = 15	`1111`
+1	1 + 8 = 9	`1001`
0	0 + 8 = 8	`1000`
-1	-1 + 8 = 7	`0111`
-8 (En küçük)	-8 + 8 = 0	`0000`

1.7 Kesirleri Depolama

Tam sayıları depolamanın aksine, bir değeri kesirli bölümü ile depolamak sadece onun ikili gösterimini temsil eden 0 ve 1'lerin desenini değil aynı zamanda taban noktasını da depolamak gerekmektedir.Bunu yapmanın bilinen yolundan biri de bilimsel gösterime dayanan **kayan nokta (floating point) gösterimidir.**

1.7.1 Kayan Nokta (Floating Point) Gösterimi

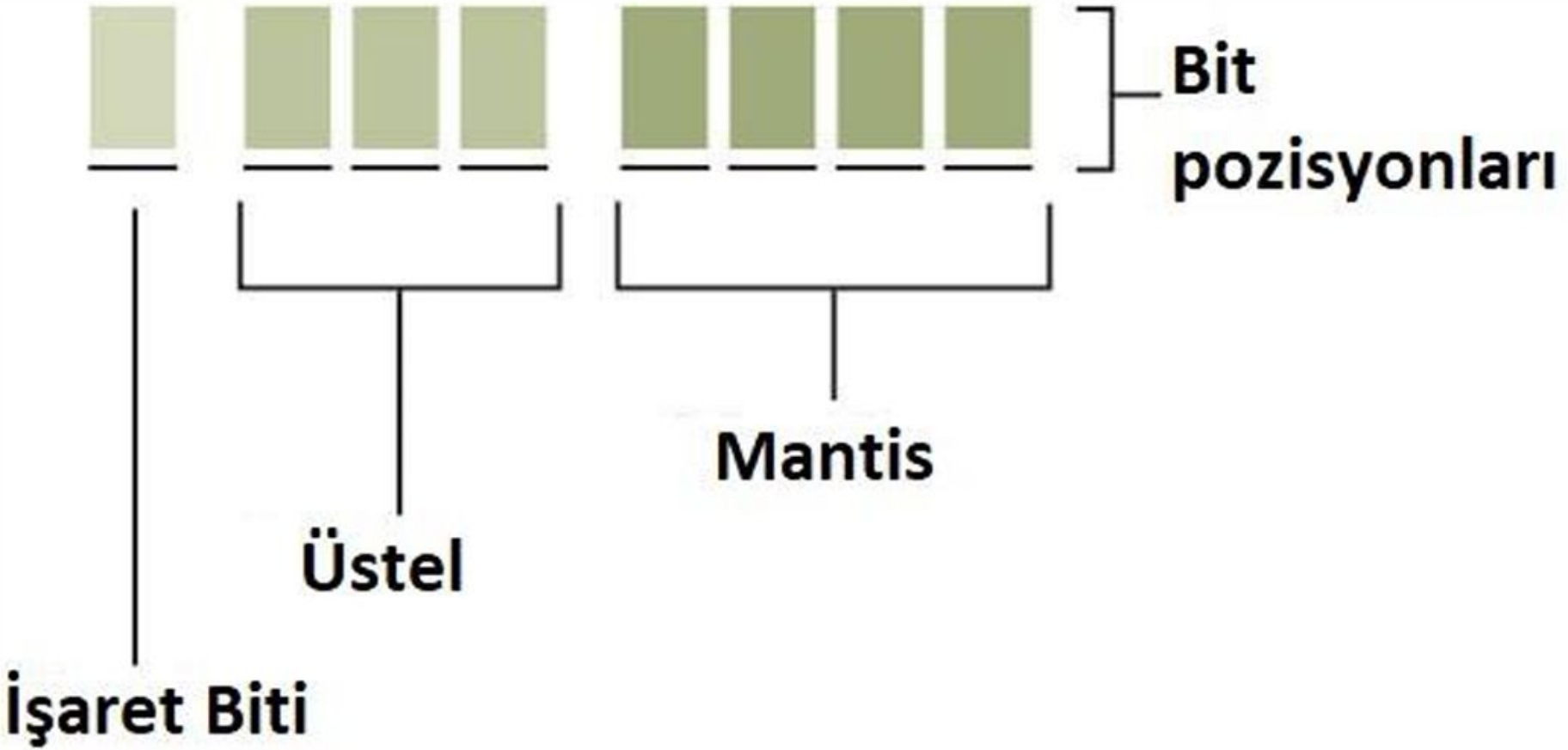
1. **İşaret Biti (Sign Bit):** Sayının pozitif mi yoksa negatif mi olduğunu belirleyen tek bir bittir.

   0: pozitif (+)

   1: negatif (-)

2. Üst / Üstel Kısım (Exponent): Sayının büyüklüğünü (ölçeğini) belirler. Virgülün (noktanın) ne kadar sağa veya sola kayacağını kontrol eder.

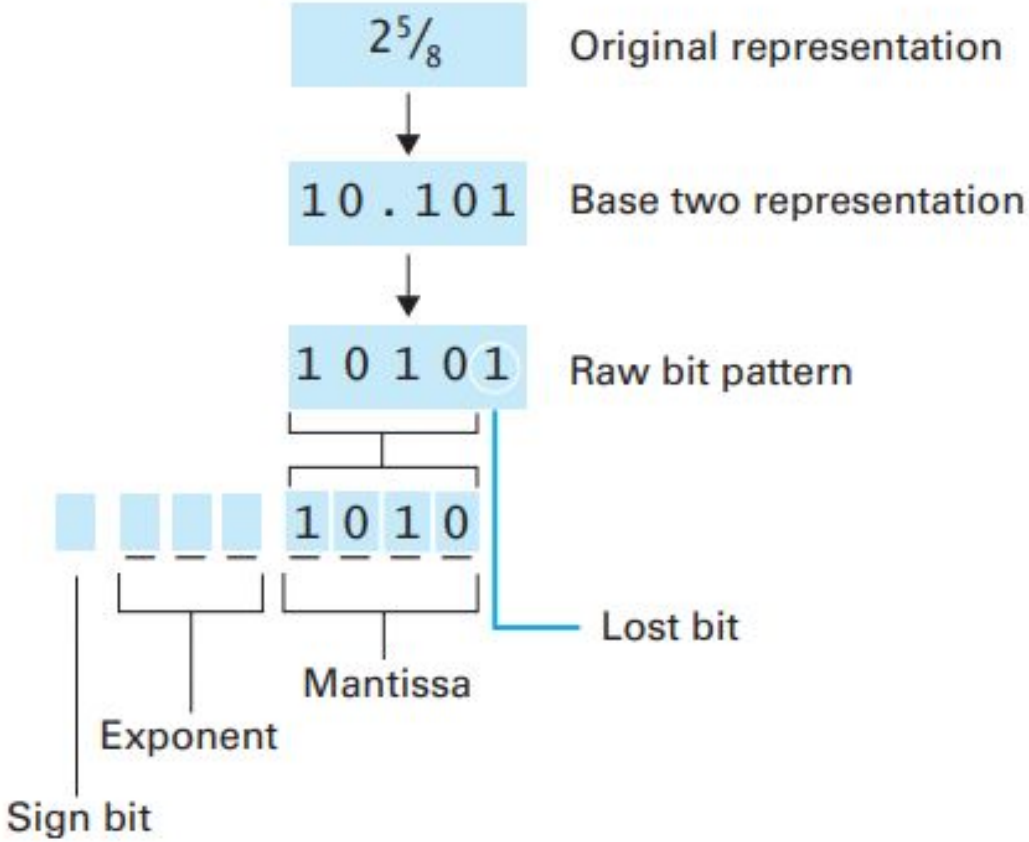
3. Mantis / Anlamlı Kısım (Mantissa / Fraction): Sayının rakamlarını (hassasiyetini) saklayan kısımdır. Bilimsel gösterimdeki ana sayıyı temsil eder.


Bit
pozisyonları
Mantis
Üstel
İşaret Biti

1.7.2 Kırpma (Yuvarlama) Hataları

- Mantis yeterince büyük olmadığı için depolanan verinin bir kısmının kaybolması durumunda ortaya çıkar

- Sonsuz kesir açılımları
  - Daha çok ikilik sayı gösteriminde görülür
  - Bu değerler genellikle tam sayılara dönüştürülür


2 5/8
Original representation
10.101
Base two representation
10101
Raw bit pattern
1010
Lost bit
Mantissa
Exponent
Sign bit

1.8 Veri ve Programlama

Bir programlama dili, insanların yüksek seviyede soyutlama kullanarak bilgisayarlara algoritmaları tam olarak ifade etmesini sağlamak için oluşturulmuş bilgisayar sistemleridir.

1.8.1 Python ile Programlamaya Giriş

Python, Guido van Rossum tarafından 1980'lerin sonunda geliştirilmiş bir programlama dilidir. Günümüzde:

- Web uygulamaları
- Bilimsel hesaplamalar
- Eğitim ve öğretim alanlarında yaygın olarak kullanılmaktadır.

Python;

- Okunabilirliğe önem verir.
- Öğrenmesi görece kolaydır.
- Zorunlu (imperative), nesne yönelimli ve fonksiyonel programlama yaklaşımlarını destekler.

Python yazılımları ücretsizdir ve python.org üzerinden erişilebilir

Python **yorumlanan (interpreted)** bir dildir. Yani:

- Komutlar doğrudan etkileşimli ortamda çalıştırılabilir
- Ya da bir dosyaya (script) yazılıp daha sonra çalıştırılabilir

1.8.1.1 İlk Program: Hello World

Giriş :

```
print('Hello, World!')
```

Çıktı :

```
Hello, World!
```

1.8.1.2 Ekrana Yazdırma

Python'da bir metni ekrana yazdırmak için `print()` fonksiyonu kullanılır. Metinler tek tırnak (`' '`) veya çift tırnak içine alınarak "string" (karakter dizisi) olarak tanımlanır. Python, tırnak içindeki ifadeyi olduğu gibi kullanıcıya geri döndürür.

1.8.1.3 Değişken ve İsimlendirme Kuralları

Değişken Atama: `=` sembolü matematikteki eşitlikten farklı olarak "atama" anlamına gelir. Değer soldaki isme atanır.

Dinamik Tip: Python'da değişkenin türünü (sayı, metin vb.) önceden belirtmeye gerek yoktur; dil bunu otomatik anlar.

Kurallar: Değişken isimleri harfle başlamalıdır; sayı, harf ve alt çizgi (`_`) içerebilir. Python **büyük-küçük harfe duyarlıdır** (`size` ve `Size` farklıdır).

Veri Tipleri: Metinler için `str`, tam sayılar için `int`, ondalıklı sayılar için `float` ve mantıksal değerler için `bool` kullanılır.

1.8.1.4 Operatörler ve Yorum Satırları

Matematiksel İşlemler: Toplama (+), çıkarma (-), çarpma (*), bölme (/) ve üs alma (**) gibi operatörler kullanılır. // tam sayı bölmesi yaparken, % kalanını (modül) verir.

Yorumlar: # sembolü ile başlayan açıklamalar bilgisayar tarafından görmezden gelinir; sadece insanlar içindir.

1.8.1.5 Hata Türleri (Debugging)

Sözdizimi Hataları (Syntax Errors): Yazım kurallarına uymama (örneğin parantezi kapatmamak).

Anlamsal Hatalar (Semantic Errors): Kod çalışır ama yanlış sonuç verir (işlem önceliği hatası gibi).

Çalışma Zamanı Hataları (Runtime Errors): Program çalışırken oluşan hatalar (sıfıra bölme gibi).

1.9 Veri Sıkıştırma Teknikleri

Verilerin daha az yer kaplaması için kullanılan kayıplı ve kayıpsız yöntemleri açıklayacağız.

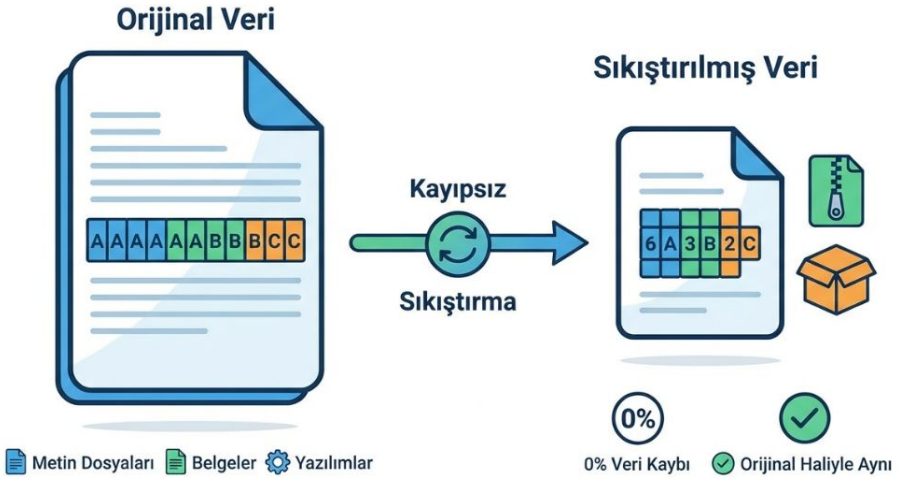
1- Kayıpsız Sıkıştırma Yöntemleri

-Run-Length Encoding (RLE)

-Sıklığa Bağlı Kodlama (Huffman)

-Sözlük Kodlaması (LZW)

- **Run-Length Encoding (RLE):** Tekrar eden veri dizilerini (örneğin 100 tane "1") sayısal bir özetle belirtir.
- **Sıklığa Bağlı Kodlama (Huffman):**En çok kullanılan öğelere kısa, az kullanılanlara uzun kodlar vererek yer tasarrufu sağlar.
- **Sözlük Kodlaması (LZW):**Veri içindeki kelime veya kalıpları bir sözlüğe kaydeder ve tekrar eden kalıpları sözlük numarasıyla çağırır.



2-Ses ve Video Sıkıştırma (MPEG, MP3)

- MPEG (Video): Videoyu resim kareleri (I-frames) olarak görür ve sadece kareler arasındaki farkları kaydederek (bağıl kodlama) devasa boyutları küçültür.

- MP3 (Ses): İnsan kulağının duyamayacağı frekansları ve yüksek seslerin gölgelediği zayıf sesleri siler (işitsel maskeleme).

Python Projesi: “RLE (Run-Length Encoding) Sıkıştırıcı”

• Açıklama: Basit bir veriyi (örneğin ardışık tekrarlayan karakterlerden oluşan bir metni veya 0/1 matrisini) sıkıştırarak boyutunu azaltan bir araç.
• Kod Beklentisi: Kullanıcı “AAAAABBBCCDAA” gibi bir girdi verdiğinde bunu “5A3B2C1D2A” haline getiren (encode) ve tam tersini yapıp eski haline döndüren (decode) iki fonksiyonlu bir program. Sıkıştırma oranını (%) hesaplamalıdır.

Hazırlayan: Gülşen Çeken

Öğrenci No: 25360859006

İletişim: gulsenceken13@gmail.com

VAKİT AYIRDIĞINIZ İÇİN TEŞEKKÜR EDERİM … •◡•