Bölüm 5: Dizgi Algoritmaları

Algoritmalar

Dizgi Algoritmaları

- Metinlerle dolu bir dünyada yaşıyoruz.
- E-postalar, mesajlar, sosyal medya paylaşımları, haber metinleri...
- Bilgisayarlarımızda her gün sayısız metinle karşılaşıyoruz.
- Peki, bu metinler nasıl düzenlenir ve analiz edilir?
- Dizgi (String) algoritmaları,
  - metinlerde arama,
  - değiştirme,
  - karşılaştırma gibi işlemleri gerçekleştirir.

Dizgi Eşleştirme Algoritmaları

- Brute Force (Kaba Kuvvet):
  - Metindeki her konum örüntü ile eşleştirmek için kontrol edilir.
  - Maksimum sayıda karşılaştırma gerektirebilir.
- Knuth-Morris-Pratt (KMP)
  - Başlangıçta tablo oluşturularak arama süresi azaltılır,
  - Karakter karşılaştırmalarını azaltarak hızlı çalışır.
- Boyer-Moore
  - Uzun aramalarda etkili. Kök bulma ve kaydırma stratejisi kullanır.
- Rabin-Karp Algoritması
  - Olasılıksal bir algoritma. Hashing kullanır.

Dizgi Sıkıştırma Algoritmaları

- Sıralı Sıkıştırma Kodlaması (Run Length Encoding)
  - Aynı veri değerleri tek bir değer ve sayı olarak saklanır.
  - Tekrar eden değerler yerine tekrar eden veri sayısı saklanır.
- Lempel-Ziv-Welch (LZW)
  - GIF gibi formatlarda kullanılan sözlük tabanlı sıkıştırma algoritması.
  - Tekrar eden örüntüleri sözlük oluşturarak kısa sembollerle temsil eder.
  - Dinamik bir sözlük kullanarak sıkıştırma sağlar.

Dizgi Ayrıştırma (Parsing) Algoritmaları

- Düzenli İfadeler (Regular Expressions)
  - Bir arama örüntüsünü tanımlayan karakter dizisi,
  - Belirli bir örüntüye uyan tüm dizgileri bulmak için kullanılır
- Sonlu Durum Makineleri (Finite State Machines - FSM)
  - Dizgi içindeki örüntüleri tanımak için kullanılan hesaplama modelleri,
  - Belirli bir girdi dizisindeki geçişlerin durumlarını izleyen bir otomat,
  - Karmaşık ayrıştırma ve analiz işlemlerinde kullanılır.

Dizgi Düzenleme Mesafesi Algoritmaları

- Levenshtein Mesafesi
  - İki dizgi arasındaki benzerliği ölçen bir metrik,
  - Bir dizgiden diğerine dönüştürmek için gereken minimum tek karakterli düzenleme sayısı olarak tanımlanır.
- En Uzun Ortak Alt Dizi (Longest Common Subsequence - LCS)
  - İki dizginin ortak olan en uzun alt dizisi,
  - Karakterlerin sıralı olmasını gerektirmez, ancak sıra korunmalıdır.
  - Dizgiler arasındaki benzerlik veya farkı belirlemek için kullanılır.

Dizgi Dönüşüm Algoritmaları

- Sonek Dizisi (Suffix Array)
  - Bir dizginin tüm son eklerinin bir dizisi.
  - Dizgi içindeki alt dizgilerin bir temsili olarak kullanılır.
- Burrows-Wheeler Dönüşümü (BWT)
  - Bir dizginin tersine dönüştürülmesiyle elde edilen yeni bir form,
  - Bzip2 gibi sıkıştırma algoritmaları için önişlem adımı olarak kullanılır.

Dizgi Sıralama Algoritmaları

- Alfabetik Sıralama (Lexicographic Order)
  - Dizgiler, alfabetik sıraya benzer sıralanır.
  - Her karakterin ASCII değeri karşılaştırılarak sıralama yapılır.
- Taban Sıralama (Radix Sort)
  - Karşılaştırmalı olmayan bir tam sayı sıralama algoritmasıdır.
  - Veriler tamsayı anahtarlarına sahiptir.
  - Aynı konumda aynı değeri paylaşan verileri gruplandırarak sıralar.
  - Her basamak için ayrı ayrı işlem yapılır.

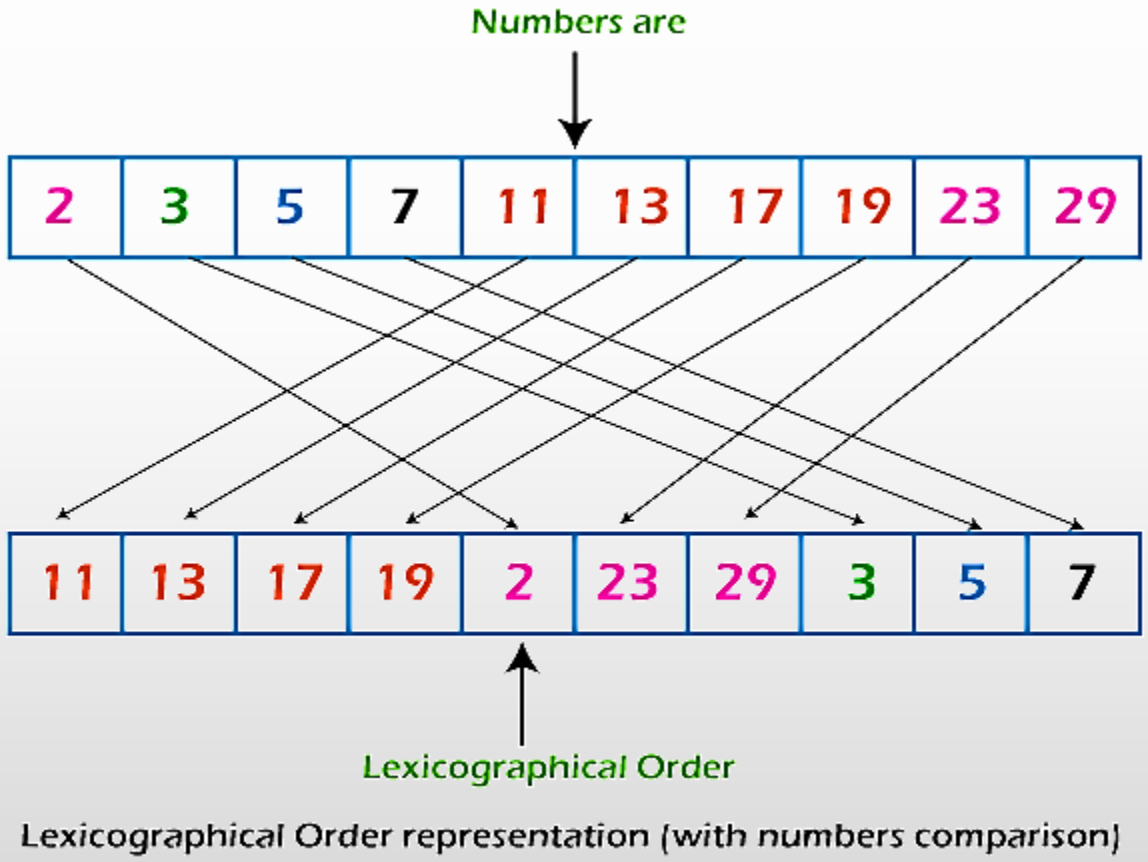
Alfabetik Sıralama

- Bir dizi öğeyi (kelimeler, sayılar, vb.) belirli bir düzene göre sıralar.
- Karakterlerin alfabedeki veya sayısal düzende pozisyonlarına dayanır.
- "apple" kelimesi "banana" kelimesinden önce gelir,
  - çünkü "a" harfi "b" harfinden önce gelir.
- 123, 45, 6, 789 gibi rakamlar, soldan sağa doğru sıralanır.

Özellikler

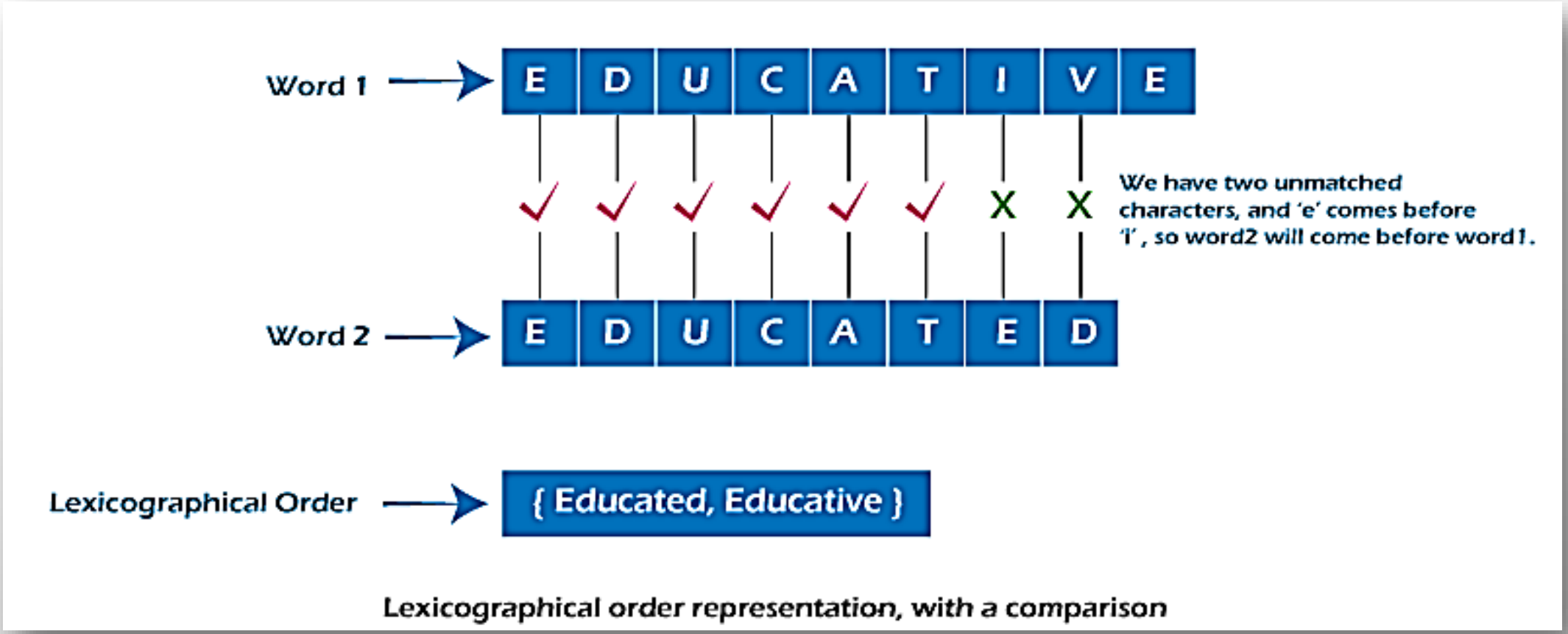
- İlk karakterlerin karşılaştırılmasıyla başlar.
- Eğer ilk karakterler eşitse, bir sonraki karakterlere bakılır.
- Bu işlem öğelerin tamamı karşılaştırılana kadar devam eder.

Alfabetik Sıralama



Lexicographical Order representation (with numbers comparison)

Alfabetik Sıralama



Lexicographical order representation, with a comparison

Örnek

- Diziler:
  - *apple* ve *apricot*
- İlk Karakterler:
  - *a* ve *a* eşit.
- İkinci Karakterler:
  - *p* ve *p* eşit.
- Üçüncü Karakterler:
  - *p* ve *r* karşılaştırılır.
  - ASCII değeri *p (112) < r (114)*,
  - *apple apricot*'tan önce gelir.

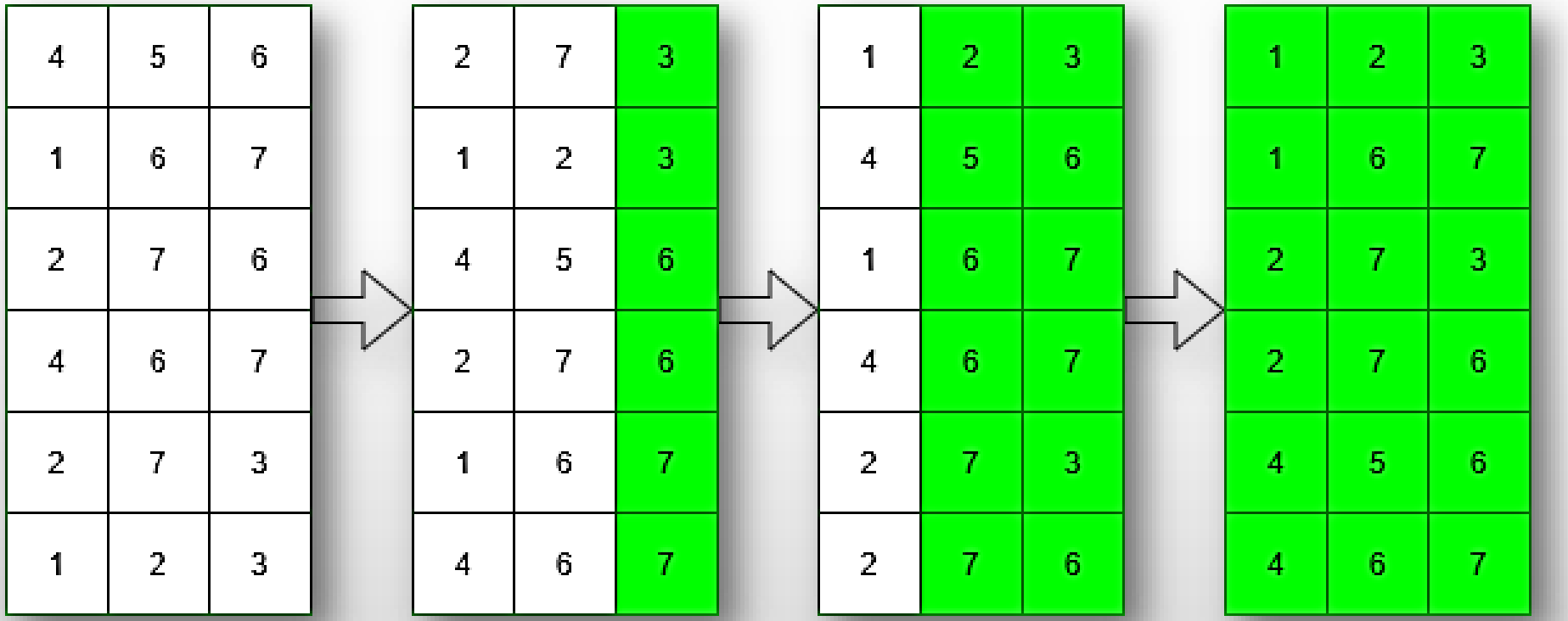
Taban Sıralama (Radix Sort)

- Her bir karakteri basamak değeri gibi kullanarak sıralar.
- En uzun dizginin uzunluğu bulunur.
- Sağdan başlayarak her bir karakter basamak olarak ele alınır.
- Sıralama işlemi en önemli basamaktan az önemli basamağa doğru yapılır.
- Her bir basamakta, öğeler alfabetik olarak sıralanır.

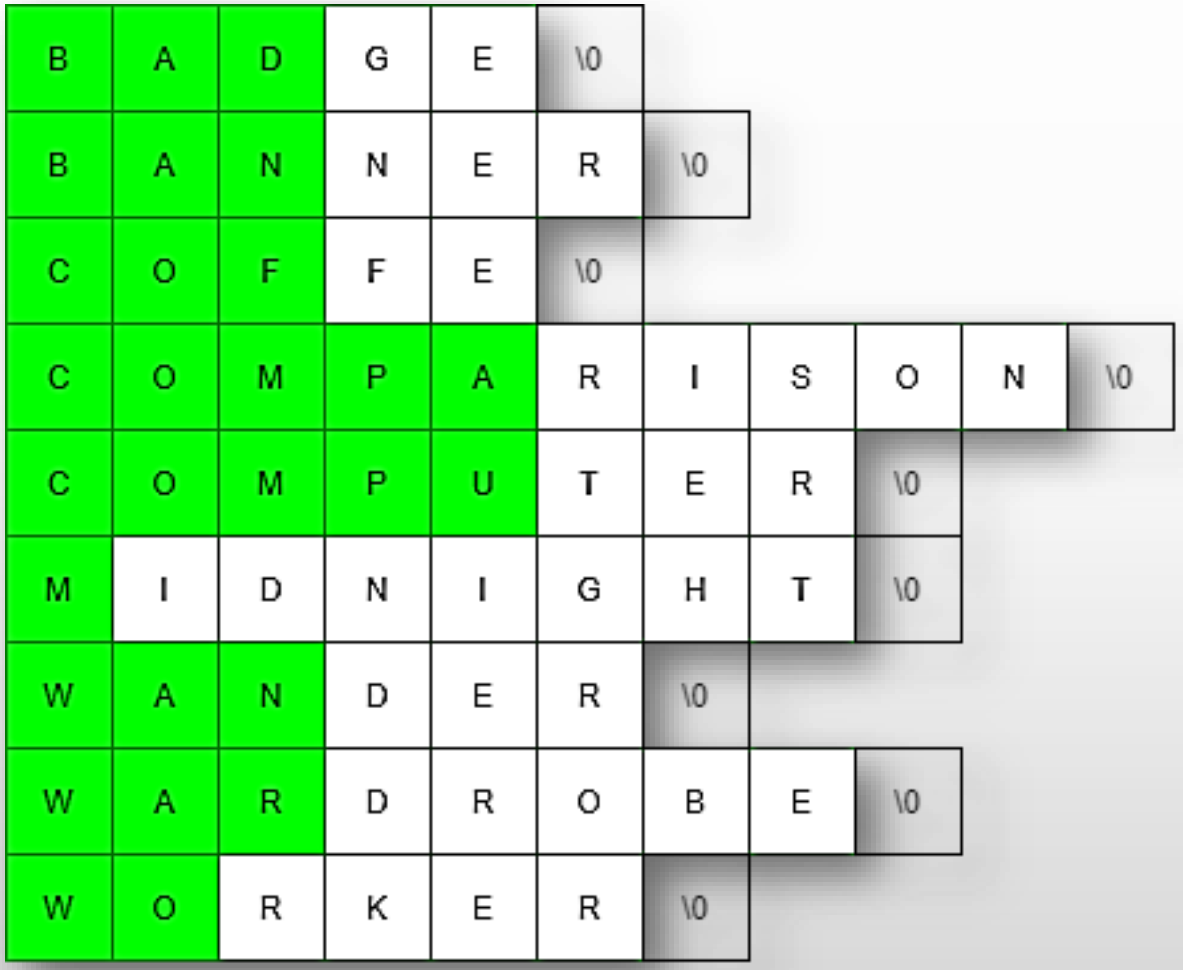
Zaman Karmaşıklığı

- En uzun dizginin uzunluğu (n) * Eleman sayısı (N)
- *O(n N)*

Radix Sort



Radix Sort



Örnek

- Veri Kümesi: [170, 45, 75, 90, 802, 24, 2, 66]
- En Az Anlamlı Basamağa Göre:
  - 170, 45, 75, 90, 802, 24, 2, 66 → (0, 5, 5, 0, 2, 4, 2, 6)
  - [170, 90, 802, 2, 24, 45, 75, 66]
- İkinci Basamağa Göre:
  - 170, 90, 802, 2, 24, 45, 75, 66 → (7, 9, 0, 0, 2, 4, 7, 6)
  - [802, 2, 24, 45, 66, 170, 75, 90]
- En Anlamlı Basamağa Göre:
  - 802, 2, 24, 45, 66, 170, 75, 90 → (8, 0, 0, 0, 0, 1, 0, 0)
  - [2, 24, 45, 66, 75, 90, 170, 802]

SON