



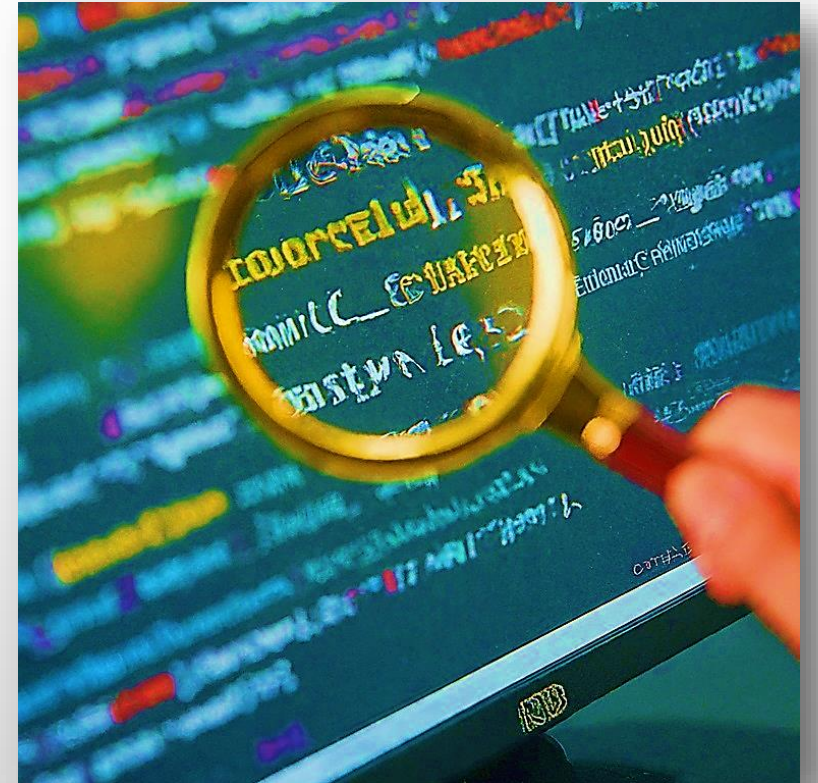
Bölüm 3: Arama Algoritmaları

Algoritmalar



Arama Algoritmaları

- Bir veri kümesinde istenilen değeri bulmak için kullanılır.
- Arama kriterine göre veriyi tarar ve
 - eşleşen öğeyi bulmaya çalışır.
- Web sitelerinde, müzik çalarlarda, akıllı telefonlarda yaygın kullanılır.





Arama Algoritmalarının Çeşitleri

- Farklı arama algoritmaları, farklı çalışma prensiplerine sahiptir.
- Doğrusal Arama (Linear Search):
 - Verileri tek tek karşılaştırarak arar.
- İkili Arama (Binary Search):
 - Veri kümesini ikiye bölerek ve arama alanını daraltarak arar.
- Hash Arama (Hash Search):
 - Veriyi hızlı erişim için bir hash tablosuna yerleştirir ve
 - Aramayı tablo üzerinden yapar.



Doğrusal Arama

- Bazen aranan bilgiyi bulmak için tek tek bakmak en iyi yoldur.
- Alışveriş listesindeki ürünleri tek tek kontrol ederek aramaya benzer.
- Karmaşık olmayan durumlarda oldukça kullanışlıdır.





Doğrusal Arama

- Listedeki her öğeyi tek tek kontrol ederek arama yapar.
- Kayıp bir eşyayı bulmak için odayı sistematik olarak tarama gibi.
- Aranan değer, listenin başından başlanarak her öge ile karşılaştırılır.
- Eğer aranan değer bulunursa, konumu döndürülür.
- Eğer aranan değer listede yoksa, başarısız sonuç döndürülür.



Avantajları

- Kodlaması ve anlaşılması kolay.
- Karmaşık veri yapıları gerektirmez.
- Ön hazırlık süreci yoktur.
- Küçük veri kümelerinde hızlı bir şekilde arama yapmak için idealdir.
- Örneğin, bir telefon numarasını küçük bir rehber listesinde aramak.



İkili Arama

- Kütüphanede kitap aramaya benzer.
- Kitaplar, yazarın soyadına göre alfabetik olarak sıralanmıştır.
- Aranan kitabı bulmak için tüm rafları tek tek aramak yerine,
 - önce orta sıradaki bölüme gidilir.
- Eğer kitap alfabetik olarak orta sıranın solundaysa,
 - sol taraftaki raflar aranmaya devam edilir.
- Sağdaysa, sağ taraftaki raflar kontrol edilir.





Avantajlar

- Önceden sıralı listelerde arama yaparken hızlıdır.
- Her adımda, listenin kontrol edilmesi gereken kısmını yarıya indirir.
- Örneğin, 1000 öğelik bir listede
 - doğrusal arama ortalama 500 kontrol yaparken,
 - ikili arama sadece 10 adımda aramayı tamamlar.
- Büyük ve sıralı veri kümelerinde arama için ideal.





Hash Tablo Arama Algoritması

- Bir anahtarın kilide tam olarak oturması gibi çalışır.
- Veriler, anahtar kelimeler ve bunlara karşılık gelen değerlerden oluşur.
- Anahtar kelimeler, hash fonksiyonu ile benzersiz değerlere dönüştürülür.
- Hash değeri, hash tablosundaki verilerin yerini işaret eder.
- Aranan anahtar kelimenin hash değeri ile konumda (kova) arama yapılır.
- Eğer kova boş değilse, anahtar kelime kovadaki değerlerle karşılaştırılır,
 - aranan değer bulunursa işlem tamamlanır.





Avantajları

- Ortalama durumda çok hızlı arama yapar.
- Verilerin önceden sıralanmasına gerek yoktur.
- Büyük veri kümeleri aramalarında idealdir.
- Hash fonksiyonu iyi seçilmişse, aranılan bilgiye doğrudan erişilebilir.
- Hash fonksiyonu çakışmalara yol açabilir.
 - iki farklı anahtar kelimenin aynı hash değerine sahip olması.
- Çakışmalar olduğunda, ek işlem adımları gerekebilir. 😞



Aradeğer (Interpolation) Arama

- Sadece önceden sıralanmış veri setlerinde kullanılabilir.
- Verilerin listenin içinde eşit aralıklarla dağıldığını varsayar.
- Listenin başlangıç değerinden sonra her eleman arasındaki fark aynıdır.
- Bu varsayıma dayanarak, aranan değerın konumu tahmin edilir.
- Tahmin edilen konum kontrol edilir ve eğer yanlışsa,
 - arama daha dar bir aralıkta (sağda veya solda) devam ettirilir.



Aradeğer (Interpolation) Arama

- Aranan değer, listenin en küçük ve en büyük elemanı kullanılarak tahmini bir konuma yerleştirilir.
- Tahmini konumdaki değer kontrol edilir.
- Eğer tahmini konumdaki değer aranan değer ise, arama başarıyla tamamlanır ve konum döndürülür.
- Eğer tahmini konumdaki değer aranan değerden küçükse, arama listenin sağ yarısında devam ettirilir. Tahmin güncellenerek yeni bir konum hesaplanır.
- Eğer tahmini konumdaki değer aranan değerden büyükse, arama listenin sol yarısında devam ettirilir. Tahmin güncellenerek yeni bir konum hesaplanır.
- Bu adımlar, aranan değer bulunana kadar veya listenin tüm elemanları kontrol edilene kadar tekrarlanır.



Atla Ara Bul (Jump) Algoritması

- Engelli koşuda engelleri aşarak ilerleyen bir atlet gibi çalışır.
- Listeyi önceden belirlenen büyüklükte parçalara ayırır.
- İlk olarak listenin başından büyük bir adım atlar ve o konumdaki öğeyi kontrol eder.
- Eğer aranan değer bu konumdaysa, arama başarıyla sona erer.
- Eğer atlanan konumdaysa, atlanan aralığa geri dönülerek o aralıkta doğrusal arama yapılır.
- Bu sayede, tek tek tüm öğeleri kontrol etmek yerine daha hızlı bir şekilde arama gerçekleştirilebilir.



Üstel (Exponential) Arama

- Define avcısının harita işaretlerini takip ederek defineye yaklaşması gibi.
- Listeyi, üstel olarak büyüyen parçalara ayırır.
- Örneğin, listenin uzunluğu 1024 ise,
 - ilk arama 1024. konuma, yani listenin sonuna yakın bir konuma yapılır.
 - aranan değer burada değilse, atlama mesafesi geriye çekilir ve
 - arama 512 ile 1023 arasındaki kısımda devam eder.
 - her adımda atlama mesafesi küçülerek arama alanı daraltılır.







SON