

Hash Tabloları ve Çarpışma Çözümü Nedir?

Çalışma Kağıdı

Hash tabloları, anahtarları bir dizi indeksine dönüştüren bir hash fonksiyonu kullanarak verileri saklar. Çarpışma çözümü ise, aynı indekse yerleştirilmesi gereken birden fazla anahtarın yönetilmesidir.

Sorular

1. Hash tablolarının temel amacı nedir?
A) Verileri sıralı tutmak
B) Elemanlara ortalama sabit zamanda erişim sağlamak
C) Bellek kullanımını optimize etmek
D) Veri bütünlüğünü sağlamak
2. Aşağıdakilerden hangisi bir çarpışma çözme yöntemi DEĞİLDİR?
A) Linear Probing
B) Quadratic Probing
C) Double Hashing
D) Binary Search Tree
3. Zincirleme (chaining) yönteminde, bir hücrede çarpışma olursa ne olur?
A) Eleman atılır.
B) Eleman tablo dışına kaydedilir.
C) Eleman, o hücredeki bağlı listenin sonuna eklenir.
D) Tablo boyutu artırılır.
4. Basit bir hash tablosu oluşturma ve eleman ekleme örneği nedir?
5. Açık adresleme (linear probing) ile çarpışma çözümü nasıl yapılır?
6. Zincirleme (chaining) ile çarpışma çözümü nasıl yapılır?
7. Tanımla: Hash Fonksiyonu Nedir?
8. Tanımla: Hash Tablosu Nedir?
9. Tanımla: Hash Çarpışması Nedir?

Cevap Anahtarı

1. B) Elemanlara ortalama sabit zamanda erişim sağlamak — Hash tabloları, hash fonksiyonu sayesinde anahtar-değer çiftlerine ortalama $O(1)$ sürede erişim sağlamak için tasarlanmıştır.
2. D) Binary Search Tree — Binary Search Tree (İkili Arama Ağacı) bir veri yapısıdır, hash çarpışması çözme yöntemi değildir. Diğer seçenekler açık adresleme yöntemleridir.
3. C) Eleman, o hücredeki bağlı listenin sonuna eklenir. — Zincirleme yönteminde, her hücre bir bağlı liste tutar. Çarpışma olduğunda, yeni eleman ilgili hücredeki listenin sonuna eklenir.
4. 1. Bir hash fonksiyonu seçin (örn. anahtarın ASCII değerlerinin toplamı mod tablo boyutu). 2. Anahtar-değer çiftlerini alın (örn. 'ali': 10, 'veli': 20). 3. Her anahtar için hash fonksiyonunu uygulayarak indeksleri hesaplayın. 4. Değerleri hesaplanan indekslere yerleştirin. 5. Çarpışma durumunda uygun çözüm yöntemini uygulayın.
5. 1. Hash fonksiyonu ile ilk indeks hesaplanır. 2. Eğer o indeks doluysa, bir sonraki boş indekse bakılır (doğrusal olarak ilerlenir). 3. Yeni eleman boş bulunan ilk indekse yerleştirilir. 4. Arama yaparken de aynı yol izlenir, eleman bulunamazsa veya boş bir hücreye ulaşırsa arama sonlandırılır.
6. 1. Her tablo hücresi bir bağlı liste başı olarak kabul edilir. 2. Bir anahtarın hash değeri hesaplanır ve ilgili hücreye gidilir. 3. Eğer hücrede zaten bir liste varsa, yeni eleman bu listenin sonuna eklenir. 4. Arama yaparken, ilgili hücredeki listenin taranması gerekir.
7. Anahtarları sabit boyutlu bir aralıktaki indekslere dönüştüren fonksiyondur.
8. Anahtar-değer çiftlerini saklamak için hash fonksiyonu kullanan bir veri yapısıdır.
9. Farklı anahtarların aynı hash değerine veya tablo indeksine eşlenmesidir.

Bounlu

Tüm kartlar, adım adım çözümler ve AI hoca desteği Notek uygulamasında.
Sınav tarihlerini Promy otomatik hatırlatıcıya çevirir.