

Karma Tablolar ve Karma Fonksiyonları Nedir?

Çalışma Kağıdı

Karma tablolar, anahtar değerlerini önceden belirlenmiş bir dizi (array) içine dağıtmak için karma fonksiyonlarını kullanır. Karma fonksiyonu, bir anahtarı alır ve bu anahtarın tablo içindeki indeksini döndürür.

Sorular

1. Karma tabloların en büyük avantajı aşağıdakilerden hangisidir?

- A) Yüksek bellek kullanımı
- B) Ortalama $O(1)$ erişim süresi
- C) Sıralı veri saklama
- D) Karmaşık uygulama gereksinimi

2. Bir karma fonksiyonunun görevi nedir?

- A) Verileri sıralamak
- B) Anahtarları benzersiz hale getirmek
- C) Anahtarları tablo indekslerine dönüştürmek
- D) Bellek kullanımını optimize etmek

3. Aşağıdakilerden hangisi bir çarpışma çözüm tekniği DEĞİLDİR?

- A) Ayrık Zincirleme
- B) Açık Adresleme
- C) Doğrusal Tarama (Linear Probing)
- D) Yığın (Stack)

4. Basit bir karma tabloya 'elma': 5, 'muz': 3, 'kiraz': 7 değerlerini ekleyelim. Modulo (kalan) alma yöntemiyle basit bir karma fonksiyonu kullanalım. Tablo boyutu 10 olsun.

5. Karma tabloya 'portakal': 9 değerini eklerken, karma fonksiyonunun 'elma' ile aynı indeksi döndürmesi durumunda ne olur? (Çarpışma/Collision)

6. Tanımla: Karma Tablo (Hash Table) Nedir?

7. Tanımla: Karma Fonksiyonu (Hash Function) Nedir?

8. Tanımla: Karma Tabloların Avantajı Nedir?

Cevap Anahtarı

1. B) Ortalama $O(1)$ erişim süresi — Karma tablolar, anahtar-değer çiftlerine ortalama sabit zamanda ($O(1)$) erişim sağlayarak büyük veri kümelerinde önemli bir performans artışı sunar.
2. C) Anahtarları tablo indekslerine dönüştürmek — Karma fonksiyonu, bir anahtar girdisi olarak bu anahtarın karma tablosundaki konumunu belirleyen bir indeks değeri üretir.
3. D) Yığın (Stack) — Yığın (Stack), LIFO (Son Giren İlk Çıkar) prensibiyle çalışan bir veri yapısıdır ve karma tablo çarpışma çözümüyle doğrudan ilgili değildir. Diğer seçenekler yaygın çarpışma çözüm teknikleridir.
4. 1. 'elma' için $karma(elma) = (e'nin\ ascii\ değeri + l'nin\ ascii\ değeri + ...) \% 10 = indeks$ 2. 'muz' için $karma(muz) = (m'nin\ ascii\ değeri + u'nun\ ascii\ değeri + ...) \% 10 = indeks$ 3. 'kiraz' için $karma(kiraz) = (k'nin\ ascii\ değeri + i'nin\ ascii\ değeri + ...) \% 10 = indeks$ 4. Elde edilen indekslere karşılık gelen değerleri tabloya yerleştiririz.
5. 1. Eğer $karma('portakal') == karma('elma')$ ise bir çarpışma meydana gelir. 2. Çarpışmayı çözmek için 'ayrık zincirleme' (separate chaining) veya 'açık adresleme' (open addressing) gibi yöntemler kullanılır. Ayrık zincirlemede, aynı indekse sahip elemanlar bir bağlı liste içinde saklanır. Açık adreslemede ise, boş bir sonraki yuva aranır.
6. Anahtar-değer çiftlerini verimli bir şekilde saklayan ve erişen veri yapısıdır.
7. Bir anahtarı alır ve tablo içindeki bir indeksi döndüren fonksiyondur.
8. Ortalama $O(1)$ zaman karmaşıklığında ekleme, silme ve arama işlemleri sunar.

Bounlu

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