

Sıralama Algoritmaları: Merge, Quick, Heap Nedir?

Çalışma Kağıdı

Merge Sort, veriyi bölerek sıralayan 'divide and conquer' prensibini kullanır; Quick Sort, pivot seçimiyle veriyi bölümleyen ve özylenelemeli çalışan bir algoritmadır; Heap Sort ise yığın (heap) veri yapısını kullanarak sıralama yapar.

Sorular

1. Aşağıdakilerden hangisi 'divide and conquer' stratejisini kullanır?

- A) Heap Sort
- B) Bubble Sort
- C) Merge Sort
- D) Insertion Sort

2. Quick Sort'un en kötü durum zaman karmaşıklığı nedir?

- A) $O(n)$
- B) $O(n \log n)$
- C) $O(n^2)$
- D) $O(\log n)$

3. Merge Sort'un ekstra bellek ihtiyacı genellikle nasıldır?

- A) Sabit $O(1)$
- B) Lineer $O(n)$
- C) Logaritmik $O(\log n)$
- D) Karesel $O(n^2)$

4. Merge Sort ile [38, 27, 43, 3, 9, 82, 10] dizisini sıralayın.

5. Quick Sort ile [10, 7, 8, 9, 1, 5] dizisini sıralayın. Pivot olarak son elemanı seçelim.

6. Heap Sort ile [4, 10, 3, 5, 1] dizisini sıralayın.

7. Tanımla: Merge Sort'un zaman karmaşıklığı nedir?

8. Tanımla: Quick Sort'un ortalama zaman karmaşıklığı nedir?

9. Tanımla: Heap Sort'un en kötü durum zaman karmaşıklığı nedir?

Cevap Anahtarı

1. C) Merge Sort — Merge Sort, problemi alt problemlere bölerek çözen bir 'divide and conquer' algoritmasıdır.
2. C) $O(n^2)$ — Quick Sort'ta pivot seçimi kötü yapıldığında (her zaman en küçük veya en büyük eleman seçildiğinde) zaman karmaşıklığı $O(n^2)$ olur.
3. B) Linear $O(n)$ — Merge Sort, birleştirme işlemi sırasında geçici olarak orijinal dizi kadar ($O(n)$) ek bellek alanına ihtiyaç duyar.
4. 1. Diziyi sürekli ikiye bölerek tek elemanlı alt dizilere ayırın. 2. Tek elemanlı alt dizileri birleştirirken sıralı bir şekilde birleştirin. 3. Tüm dizi birleşene kadar bu işleme devam edin. Sonuç: [3, 9, 10, 27, 38, 43, 82]
5. 1. Dizinin son elemanını (5) pivot olarak seçin. 2. Diziyi pivot'tan küçükler ve büyükler olarak ikiye ayırın: [1, 5, 8, 9, 7, 10] 3. Pivot'un solundaki [1] ve sağındaki [8, 9, 7, 10] alt dizileri için aynı işlemi özyinelemeli olarak uygulayın. 4. Tüm alt diziler sıralandığında birleştirin. Sonuç: [1, 5, 7, 8, 9, 10]
6. 1. Diziyi bir max-heap yapısına dönüştürün: [10, 5, 3, 4, 1] 2. En büyük elemanı (heap'in kökü, 10) dizinin sonuna taşıyın ve heap'ten çıkarın. Dizin boyutu 1 azaltılır. 3. Kalan elemanlarla tekrar heap özelliğini sağlayın ve en büyük elemanı (5) bir önceki elemanın yanına taşıyın. 4. Tüm elemanlar sıralanana kadar tekrarlayın. Sonuç: [1, 3, 4, 5, 10]
7. Her durumda $O(n \log n)$ 'dir.
8. $O(n \log n)$ 'dir.
9. $O(n \log n)$ 'dir.

Bounlu

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