Heapsort Algoritması Nedir?

Çalışma Kağıdı

Heapsort, bir diziyi iki aşamada sıralar: önce diziyi bir yığına dönüştürür, ardından yığından en büyük (veya en küçük) elemanı çıkararak sıralı bir dizi oluşturur.

Sorular

1. Heapsort algoritmasının ilk adımı nedir?

A) Diziyi yığına dönüştürmek
B) Diziyi ikiye bölmek
C) En küçük elemanı bulmak
D) Diziyi ters çevirmek

2. Aşağıdakilerden hangisi Heapsort'un avantajlarından biridir?

A) Kararlılık
B) Düşük yer karmaşıklığı (O(1))
C) Her zaman O(n) zaman karmaşıklığı
D) Daha az karşılaştırma yapması

3. Heapsort'un 'heapify' işlemi neyi sağlar?

A) Diziyi tersine çevirmeyi
B) Yığın özelliğini korumayı
C) Elemanları rastgele yerleştirmeyi
D) Diziyi ikiye bölmeyi

4. Heapsort ile [4, 10, 3, 5, 1] dizisini sıralayın.

5. Heapsort'un 'heapify' işlemi nedir?

6. Tanımla: Heapsort hangi veri yapısını kullanır?

7. Tanımla: Heapsort'un zaman karmaşıklığı nedir (ortalama)?

8. Tanımla: Heapsort'un zaman karmaşıklığı nedir (en kötü)?

Cevap Anahtarı

1. A) Diziyi yığına dönüştürmek — Heapsort'un ilk adımı, verilen diziyi bir yığın (genellikle maksimum yığın) veri yapısına dönüştürmektir.

2. B) Düşük yer karmaşıklığı (O(1)) — Heapsort, sıralamayı orijinal dizi üzerinde yaptığı için O(1) yer karmaşıklığına sahiptir, yani yerinde bir sıralama algoritmasıdır.

3. B) Yığın özelliğini korumayı — Heapify işlemi, bir düğümün ve alt ağaçlarının yığın (heap) özelliğini korumasını sağlar.

4. 1. Yığın Oluşturma: Diziyi maksimum yığına dönüştürün. [10, 5, 3, 4, 1] 2. Sıralama: - En büyük eleman (10) dizinin sonuna taşınır. Dizi: [1, 5, 3, 4], Yığın: [10] - Kalan elemanlarla yığın tekrar düzenlenir: [5, 4, 3, 1] - En büyük eleman (5) son elemanın önüne taşınır. Dizi: [1, 4, 3], Yığın: [5, 10] - Kalan elemanlarla yığın tekrar düzenlenir: [4, 1, 3] - En büyük eleman (4) son elemanın önüne taşınır. Dizi: [1, 3], Yığın: [4, 5, 10] - Kalan elemanlarla yığın tekrar düzenlenir: [3, 1] - En büyük eleman (3) son elemanın önüne taşınır. Dizi: [1], Yığın: [3, 4, 5, 10] - Son eleman (1) en başa konulur. Dizi: [1], Yığın: [1, 3, 4, 5, 10] Sonuç: [1, 3, 4, 5, 10]

5. Heapify, bir düğümün ve onun alt ağaçlarının yığın özelliğini korumasını sağlayan bir işlemdir. Bir düğümün altındaki elemanlar yığın özelliğini bozuyorsa, düğüm ile altındaki en büyük çocuk yer değiştirilir ve işlem özyinelemeli olarak devam eder.

6. Yığın (Heap)

7. O(n log n)

8. O(n log n)