

Dizi Yapıları

1) Bir dizinin avantajları arasında hangisi değildir?

- a) Hızlı eleman erişimi
- b) Esnek boyut
- c) Veri saklama sınırlamaları
- d) Bellek verimliliği

2) Verilen bir diziyi tersine çevirmek için kullanılan algoritma hangisidir?

- a) Merge Sort
- b) Bubble Sort
- c) Reverse Algorithm
- d) Quick Sort

3) Dizi elemanlarına erişim için kullanılan indeksleme avantajları nelerdir?

- a) $O(1)$ zaman karmaşıklığı
- b) Dinamik boyut
- c) Sıralı erişim
- d) Heap bellek kullanımı

4) Bir dizideki elemanların toplamını bulan algoritma hangisidir?

- a) Linear Search
- b) Binary Search
- c) Merge Sort
- d) Sum Algorithm

5) Dizideki en büyük elemanı bulan algoritma aşağıdakilerden hangisidir?

- a) Bubble Sort
- b) Quick Sort
- c) Max Algorithm
- d) Binary Search

Liste Yapısı

6) Bağlı liste ve dizinin avantajlarını karşılaştırın.

- a) Dizi, bağlı listeye göre daha hızlıdır.
- b) Bağlı liste, dizinin bellek kullanımında daha etkilidir.
- c) Dizide eleman silme işlemi daha kolaydır.
- d) Bağlı liste, dizideki indeksleme avantajlarına sahiptir.

7) Çift yönlü bağlı listelerin kullanım alanları nelerdir?

- a) Geri alma (undo) işlemleri
- b) Hızlı eleman erişimi
- c) İndeksleme avantajları
- d) Bellek verimliliği

8) Bir bağlı listenin başına eleman eklemek için kullanılan algoritma nedir?

- a) Insertion Sort
- b) Quick Sort
- c) Linked List
- d) Prepend Algorithm

9) Bağlı listenin ortasından eleman silme işlemi için kullanılan algoritma hangisidir?

- a) Delete Middle Element Algorithm
- b) Merge Sort
- c) Binary Search
- d) Bubble Sort

10) Bağlı listenin sonundan eleman silmek için kullanılan algoritma nedir?

- a) Delete Last Element Algorithm
- b) Reverse Algorithm
- c) Insertion Sort
- d) Linear Search

Bağlı Liste Yapısı:

11) Bağlı listenin başına eleman eklemek için kullanılan algoritmanın zaman karmaşıklığı nedir?

- a) $O(1)$
- b) $O(\log n)$
- c) $O(n)$
- d) $O(n^2)$

12) Bağlı listenin sonuna eleman eklemek için kullanılan algoritma hangisidir?

- a) Append Algorithm
- b) Merge Sort
- c) Insertion Sort
- d) Bubble Sort

13) İki bağlı listeyi birleştirmek için kullanılan algoritma nedir?

- a) Connect Lists Algorithm
- b) Merge Sort
- c) Concatenate Algorithm
- d) Quick Sort

14) Bağlı listelerde döngü tespiti için kullanılan algoritma hangisidir?

- a) Circular List Algorithm
- b) Detect Loop Algorithm
- c) Reverse Algorithm
- d) Binary Search

15) Bağlı listenin belirli bir elemanını aramak için kullanılan algoritma nedir?

- a) Search Element Algorithm
- b) Binary Search
- c) Linear Search
- d) Find Node Algorithm

Sıralı Bağlı Liste Yapısı:

16) Sıralı bağlı listelerin arama işlemlerinde hangi avantaj vardır?

- a) $O(1)$ zaman karmaşıklığı
- b) Sıralı erişim
- c) Dizilere göre daha yavaştır
- d) Bellek verimliliği

17) Sıralı bağlı listelerde eleman ekleme işlemi için kullanılan algoritma hangisidir?

- a) Insertion Sort
- b) Bubble Sort
- c) Sıralı Insertion Algorithm
- d) Quick Sort

18) Sıralı bağlı listenin sonundan eleman silmek için kullanılan algoritma nedir?

- a) Delete Last Element Algorithm
- b) Remove End Element Algorithm
- c) Linear Search
- d) Binary Search

19) Sıralı bağlı listelerin avantajları arasında hangisi değildir?

- a) Hızlı arama
- b) Sıralı erişim
- c) Dinamik boyut
- d) Bellek kullanım sınırlamaları

20) İki sıralı bağlı listeyi birleştirmek için kullanılan algoritma nedir?

- a) Concatenate Algorithm
- b) Merge Sort
- c) Connect Lists Algorithm
- d) Sıralı Merge Algorithm

Yığıt Yapısı:

21) Bir yığının pop işlemi ne anlama gelir?

- a) Yığıttan eleman çıkarma
- b) Yığıta eleman ekleme
- c) Yığının en altındaki elemanı alma
- d) Yığının en üstündeki elemanı alma

22) Yığının push işlemi ne anlama gelir?

- a) Yığıttan eleman çıkarma
- b) Yığıta eleman ekleme
- c) Yığının en altındaki elemanı alma
- d) Yığının en üstündeki elemanı alma

23) Yığının LIFO özelliği nedir?

- a) First In, Last Out
- b) Last In, First Out
- c) First In, First Out
- d) Last In, Last Out

24) İki yığıtı birleştirmek için kullanılan algoritma hangisidir?

- a) Concatenate Algorithm
- b) Merge Sort
- c) Connect Stacks Algorithm
- d) Quick Sort

25) Yığıt yapısının kullanım alanları arasında hangisi bulunmaz?

- a) İşlem çağrıları (function calls)
- b) Geri alma (undo) işlemleri
- c) Derleme (compilation) işlemleri
- d) Grafik işlemleri

Kuyruk Yapısı:

26) Kuyruğun FIFO özelliği nedir?

- a) First In, Last Out
- b) Last In, First Out
- c) First In, First Out
- d) Last In, Last Out

27) Kuyruk yapısında eleman ekleme işlemine ne denir?

- a) Dequeue
- b) Enqueue
- c) Push
- d) Pop

28) Kuyruk yapısında eleman çıkarma işlemine ne denir?

- a) Dequeue
- b) Enqueue
- c) Push
- d) Pop

29) İki kuyruğu birleştirmek için kullanılan algoritma hangisidir?

- a) Concatenate Algorithm
- b) Merge Sort
- c) Connect Queues Algorithm
- d) Quick Sort

30) Kuyruk yapısının kullanım alanları arasında hangisi bulunmaz?

- a) İşlem çağrıları (function calls)
- b) İnternet sıralama algoritmaları
- c) Yazıcı kuyrukları
- d) Disk I/O işlemleri

Sıralama Algoritmaları:

31) Bubble Sort algoritmasının zaman karmaşıklığı nedir?

- a) $O(n)$
- b) $O(n \log n)$
- c) $O(n^2)$
- d) $O(\log n)$

32) Quick Sort algoritması hangi sıralama algoritmasına dayanmaktadır?

- a) Bubble Sort
- b) Insertion Sort
- c) Merge Sort
- d) Selection Sort

33) Merge Sort algoritmasının avantajları arasında hangisi yer alır?

- a) En iyi durumda $O(n)$ zaman karmaşıklığı
- b) İn-place sıralama
- c) Stabil sıralama
- d) Bubble Sort'dan daha hızlı

34) Insertion Sort algoritmasının en iyi durumda zaman karmaşıklığı nedir?

- a) $O(n)$
- b) $O(n \log n)$
- c) $O(n^2)$
- d) $O(\log n)$

35) Sıralama algoritmalarının karşılaştırma temelli olmayan bir örneği hangisidir?

- a) Quick Sort
- b) Bubble Sort
- c) Counting Sort
- d) Merge Sort

Arama Algoritmaları:

36) Binary Search algoritmasının uygulanabilmesi için ne gerekir?

- a) Sıralı bir dizi
- b) Rastgele bir dizi
- c) Bağlı liste
- d) Çift yönlü bağlı liste

37) Linear Search algoritması hangi durumda daha etkilidir?

- a) Sıralı dizilerde
- b) Rastgele dizilerde
- c) Büyük veri setlerinde
- d) Küçük veri setlerinde

38) Hashing kullanarak yapılan arama işlemlerinde hangisi doğrudur?

- a) $O(n)$ zaman karmaşıklığı
- b) $O(\log n)$ zaman karmaşıklığı
- c) $O(1)$ zaman karmaşıklığı
- d) $O(n^2)$ zaman karmaşıklığı

39) Interpolation Search algoritması hangi tür veri setlerinde daha etkilidir?

- a) Sıralı veri setleri
- b) Büyük veri setleri
- c) Rastgele veri setleri
- d) Küçük veri setleri

40) Jump Search algoritması hangi tür veri setlerinde daha etkilidir?

- a) Sıralı veri setleri
- b) Büyük veri setleri
- c) Rastgele veri setleri
- d) Küçük veri setleri

Ağaç Yapısı:

41) Ağaç yapısının avantajları arasında hangisi bulunmaz?

- a) Hızlı arama
- b) İnsanların anlayabileceği bir yapı
- c) Hafif bellek kullanımı
- d) Veri organizasyonu için etkili bir yöntem

42) Ağaç yapısında bir düğümün altındaki düğümlerin sayısına ne denir?

- a) Yaprak
- b) Dal
- c) Derece
- d) Derinlik

43) Bir ağaç yapısında bir düğümün en fazla kaç çocuğu olabilir?

- a) 1
- b) 2
- c) 3
- d) Belirli bir sınırlama yok

44) İkili ağaç yapısında bir düğümün solunda bulunan alt ağacın adı nedir?

- a) Ana dal
- b) Sağ dal
- c) Sol dal
- d) Alt dal

45) İkili ağaç yapısında bir düğümün sağındaki alt ağacın adı nedir?

- a) Ana dal
- b) Sağ dal
- c) Sol dal
- d) Alt dal

İkili Ağaç Yapısı:

46) İkili ağaç yapısında bir düğümün seviyesi nedir?

- a) Düğümün altındaki düğümlerin sayısı
- b) Düğümün üstündeki düğümlerin sayısı
- c) Düğümün derinliği
- d) Düğümün ağırlığı

47) İkili ağaç yapısında bir düğümün derinliği nedir?

- a) Düğümün altındaki düğümlerin sayısı
- b) Düğümün üstündeki düğümlerin sayısı
- c) Düğümün seviyesi
- d) Düğümün ağırlığı

48) İkili ağaç yapısında bir düğümün ağırlığı nedir?

- a) Düğümün altındaki düğümlerin sayısı
- b) Düğümün üstündeki düğümlerin sayısı
- c) Düğümün seviyesi
- d) Düğümün dal sayısı

49) In-order traversal hangi sıralamayı kullanır?

- a) Pre-order
- b) Post-order
- c) Ters sıralama
- d) Sıralı sıralama

50) İkili ağaç yapısında bir düğümün sol ve sağ alt ağaçları arasındaki yükseklik farkına ne denir?

- a) Denge faktörü
- b) Derinlik
- c) Ağırlık
- d) Seviye farkı

Dengeli Arama Ağaçları, Heaps, Çizge, Hash Tablosu:

51) AVL ağaçlarında dengenin sağlanması için hangi dengelendirme faktörü kullanılır?

- a) -1, 0, 1
- b) -2, -1, 0, 1, 2
- c) 0, 1, 2
- d) -1, 0, 2

52) Bir heap yapısında en üstteki eleman hangi özelliği sağlar?

- a) En küçük elemandır.
- b) En büyük elemandır.
- c) Orta büyüklükteki elemandır.
- d) Köktür.

53) Bir çizgede bir düğümün derecesi nedir?

- a) Düğümün altındaki düğümlerin sayısı
- b) Düğümün üstündeki düğümlerin sayısı
- c) Düğümle bağlantılı olan kenar sayısı
- d) Düğümün seviyesi

54) Hash tablosundaki çakışma durumlarını çözmek için kullanılan bir yöntem nedir?

- a) Chaining
- b) Sorting
- c) Indexing
- d) Insertion

55) Hash tablosunda bir anahtarın konumunu belirlemek için kullanılan fonksiyona ne denir?

- a) Hashing fonksiyonu
- b) Search fonksiyonu
- c) Indexing fonksiyonu
- d) Sorting fonksiyonu

1. **A)** Veri saklama sınırlamaları
2. **C)** Reverse Algorithm
3. **A)** $O(1)$ zaman karmaşıklığı
4. **D)** Sum Algorithm
5. **C)** Max Algorithm
6. **B)** Bağlı liste, dizinin bellek kullanımında daha etkilidir.
7. **A)** Geri alma (undo) işlemleri
8. **A)** Insertion Sort
9. **B)** Merge Sort
10. **A)** Delete Last Element Algorithm
11. **A)** $O(1)$
12. **A)** Append Algorithm
13. **C)** Concatenate Algorithm
14. **B)** Detect Loop Algorithm
15. **C)** Linear Search
16. **A)** $O(1)$ zaman karmaşıklığı
17. **C)** Sıralı Insertion Algorithm
18. **A)** Delete Last Element Algorithm
19. **D)** Bellek kullanım sınırlamaları
20. **C)** Connect Lists Algorithm
21. **A)** Yığıttan eleman çıkarma
22. **B)** Yığıta eleman ekleme

- 23. B)** Last In, First Out
- 24. C)** Connect Stacks Algorithm
- 25. D)** Disk I/O işlemleri
- 26. C)** First In, First Out
- 27. B)** Enqueue
- 28. A)** Dequeue
- 29. C)** Connect Queues Algorithm
- 30. A)** İşlem çağrıları (function calls)
- 31. C)** $O(n^2)$
- 32. C)** Merge Sort
- 33. C)** Stabil sıralama
- 34. A)** $O(n)$
- 35. C)** Counting Sort
- 36. A)** Sıralı bir dizi
- 37. D)** Küçük veri setlerinde
- 38. C)** $O(1)$ zaman karmaşıklığı
- 39. D)** Küçük veri setlerinde
- 40. B)** Büyük veri setleri
- 41. C)** Hafif bellek kullanımı
- 42. C)** Derece
- 43. D)** Belirli bir sınırlama yok
- 44. C)** Sol dal

45. B) Sağ dal

46. C) Düğümün seviyesi

47. A) Düğümün altındaki düğümlerin sayısı

48. A) Düğümün altındaki düğümlerin sayısı

49. A) Pre-order

50. A) Denge faktörü

51. A) -1, 0, 1

52. B) En büyük elemandır.

53. C) Düğümle bağlantılı olan kenar sayısı

54. A) Chaining

55. A) Hashing fonksiyonu

Algoritma Analizi ve Hesaplama Karmaşıklığı:

Algoritmaların çalışma sürelerini ve performanslarını analiz etme sürecidir. Hesaplama karmaşıklığı, bir algoritmanın işlem adımlarının ve kaynak kullanımının matematiksel bir ölçüsüdür.

Dizi Yapıları:

Dizi, aynı türden elemanları içeren, ardışık bellek bölgelerinde depolanan bir veri yapısıdır.

Elemanlara indeksleme ile erişilir ve bu erişim $O(1)$ karmaşıklığına sahiptir.

Liste Yapısı:

Bağlı veya bağlı olmayan elemanları içeren bir veri yapısıdır.

Bağlı liste, düğümler adı verilen yapılar aracılığıyla elemanlara bağlanır.

Bağlı Liste Yapısı:

Elemanlar, düğümler aracılığıyla bağlıdır.

Her düğüm, bir veri elemanı ve bir sonraki düğümün referansını içerir.

Sıralı Bağlı Liste Yapısı:

Bağlı liste elemanları sıralı olarak tutulur.

Arama işlemleri hızlıdır, ancak ekleme ve silme işlemleri daha karmaşıktır.

Yığıt Yapısı:

Last In, First Out (LIFO) prensibiyle çalışan bir veri yapısıdır.

Pop ve push işlemleri, en üstteki elemanla ilgili işlemleri gerçekleştirir.

Kuyruk Yapısı:

First In, First Out (FIFO) prensibiyle çalışan bir veri yapısıdır.

En önemli iki işlem enqueue (ekleme) ve dequeue (çıkarma) işlemleridir.

Sıralama Algoritmaları:

Dizideki elemanları belirli bir sıraya göre düzenleme işlemleridir.

Örnek algoritmalar arasında Bubble Sort, Quick Sort ve Merge Sort bulunur.

Arama Algoritmaları:

Veri setinde belirli bir elemanı bulma işlemleridir.

Binary Search, Linear Search gibi algoritmalar örneklendirilebilir.

Ağaç Yapısı:

Dallanmış veri yapısıdır.

Bir düğüm, bir ana düğüm ve bir veya daha fazla alt düğüm içerir.

İkili Ağaç Yapısı:

Her düğüm en fazla iki çocuğa sahiptir.

İn-order, pre-order ve post-order traversal gibi gezinme yöntemleri vardır.

Dengeli Arama Ağaçları:

AVL ağaçları gibi, her düğümün alt ağaçları arasındaki yükseklik farkını dengelemeye odaklanır.

Heaps:

Öncelik kuyruğunu uygulamak için kullanılan bir ağaç yapısıdır.

Min Heap ve Max Heap gibi tipleri bulunur.

Çizge:

Düğüm ve kenarlardan oluşan bir veri yapısıdır.

İki düğüm arasındaki ilişkiyi gösterir.

Hash Tablosu:

Anahtar-değer çiftlerini depolamak için kullanılır.

Hash fonksiyonu, anahtarı bir dizine dönüştürerek veriyi hızlı erişime olanak tanır.