

1



Kuyruk veri yapısı, doğrusal olmayan veri yapıları için yardımcı bir veri yapısı olarak kullanılamaz.

false

true

< Geri

İleri >



1



Kuyruk veri yapısı, doğrusal olmayan veri yapıları için yardımcı bir veri yapısı olarak kullanılamaz.

false

true

< Geri

İleri >



2



Diziler düzenli ya da düzensiz olarak tanımlanabilir.

false

true



< Geri

İleri >

3



Tek yönlü bağlı listede liste sonundan eleman silmenin maliyeti $O(1)$ 'dir.

true

false

< Geri

İleri >

3



Tek yönlü bağlı listede liste sonundan eleman silmenin maliyeti $O(1)$ 'dir.

true

false

< Geri

İleri >

4



Aşağıdakilerden hangisi Bağlı Listelerin avantajlarından biri değildir?

Liste başına ekleme işlevi sabit zamanda gerçekleştirilir.

Liste elemanları üzerinde dolaşmanın maliyeti $O(n)$ 'dir.

Liste başından silme işlevi sabit zamanda gerçekleştirilir.

Dinamik bellek yönetimi yapılmasına olanak verir.

< Geri

İleri >

4



Aşağıdakilerden hangisi Bağlı Listelerin avantajlarından biri değildir?

Liste başına ekleme işlevi sabit zamanda gerçekleştirilir.

Liste elemanları üzerinde dolaşmanın maliyeti $O(n)$ 'dir.

Liste başından silme işlevi sabit zamanda gerçekleştirilir.

Dinamik bellek yönetimi yapılmasına olanak verir.

< Geri

İleri >

5



Diziler sabit boyutlu olarak tanımlanır.

false

true

< Geri

İleri >



6



Dizi boyutunun dinamik olarak yönetilmesi isteniyor ise ArrayList ya da Generic List kullanılabilir.

true

false

< Geri

İleri >

6



Dizi boyutunun dinamik olarak yönetilmesi isteniyor ise ArrayList ya da Generic List kullanılabilir.

true

false

< Geri

7

İleri >



Tek yönlü bağlı listede liste başından eleman silmenin maliyeti doğrusaldır.

false

true

< Geri



İleri >

7



Tek yönlü bağlı listede liste başından eleman silmenin maliyeti doğrusaldır.

false

true

< Geri

İleri >

8



Tek yönlü bağlı listede arama yapmanın maliyeti $O(1)$ 'dir.

false

true

< Geri

İleri >



8



Tek yönlü bağlı listede arama yapmanın maliyeti $O(1)$ 'dir.

 false true

< Geri

İleri >



9



Tek yönlü bağlı listede arama yapmanın maliyeti $O(n)$ 'dir.

 false true

< Geri

İleri >



9



Tek yönlü bağlı listede arama yapmanın maliyeti $O(n)$ 'dir.

 false true

< Geri

İleri >



10



Çift yönlü bağlı listede liste başına eleman eklemenin maliyeti $O(n)$ 'dir.

 false true

< Geri

İleri >

10



Çift yönlü bağlı listede listebaşına elemanın eklenmenin maliyeti $O(n)$ 'dir.

false

true

< Geri

İleri >

11



Birden fazla eleman içeren bir yiğinda, ilk elemana doğrudan erişilebilir.

true

false

< Geri

İleri >

11



Birden fazla eleman içeren bir yiğinda, ilk elemana doğrudan erişilebilir.

true

false

< Geri

İleri >

12



Bos bir yiğin üzerinde Pop() işlevi çalışırmak Underflow hatasına sebep olur.

false

true

< Geri

İleri >

12



Bos bir yigin üzerinde Pop() islevi calistirmak Underflow hatasina sebep olur.

false

true

< Geri

ileri >



13



DeQueue() islevi kuyruk veri yapisi için ana islevdir.

true

false

< Geri

ileri >



13



DeQueue() islevi kuyruk veri yapisi için ana islevdir.

true

false

< Geri

ileri >



14



struct anahtar kelimesiyle deger tipli özel veri tipleri tasarlabilir.

false

true

< Geri

ileri >

14



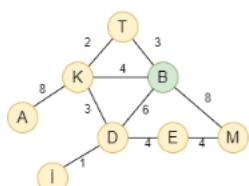
struct anahtar kelimesiyle değer tipli özel veri tipleri tasarlanabilir.

 false

 true

[« Geri](#)
[İleri »](#)

15


 Referans düğüm


Verilen çizgeyi inceleyiniz. İlgili çizgede referans düğüm yeşil renk ile vurgulanmıştır. Bu çizge üzerinde çalıştırılan bir Prims algoritmasına ilişkin aşağıdaki ifadelerden hangisi doğrudur?

 İlk ziyaret edilen düğümler (B,K)'dır ve maliyeti 4 birimdir.

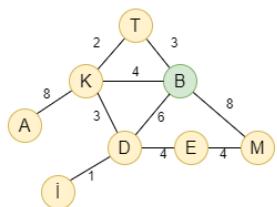
 Son ziyaret edilen düğümler (K,A)'dır.

 Başlangıç adımda gidilebilecek düğümler T ve K'dır.

 (T,K) düğümleri ziyaret edildikten sonra (K, B) düğümü ziyaret edilir.

[« Geri](#)
[İleri »](#)

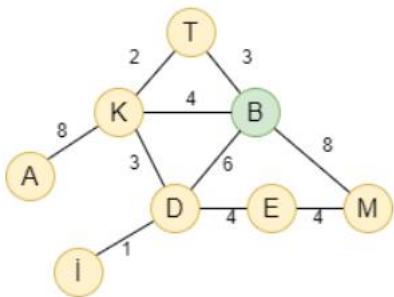
Referans düğüm



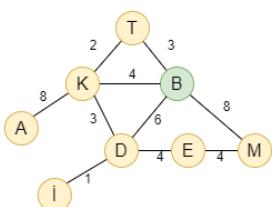
Verilen çizgeyi inceleyiniz. İlgili çizgede referans düğüm yeşil renk ile vurgulanmıştır. Bu çizge üzerinde çalıştırılan bir Prims algoritmasına ilişkin aşağıdakilerden hangisi doğrudur?

- İlk ziyaret edilen düğümler (B,K)'dır ve maliyeti 4 birimdir.
- Son ziyaret edilen düğümler (K,A)'dır.
- Başlangıç adımında gidilebilecek düğümler T ve K'dır.
- (T,K) düğümleri ziyaret edildikten sonra (K, B) düğümü ziyaret edilir.

Referans düğüm



15


 Referans düğüm


Verilen çizgeyi inceleyiniz. İlgili çizgede referans düğüm yeşil renk ile vurgulanmıştır. Bu çizge üzerinde çalıştırılan bir Prims algoritmasına ilişkin aşağıdaki ifadelerden hangisi doğrudur?

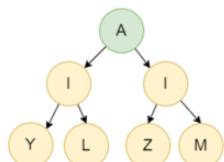
 İlk ziyaret edilen düğümler (B,K)'dır ve maliyeti 4 birimdir.

 Son ziyaret edilen düğümler (K,A)'dır.

 Başlangıç adımında gidilebilecek düğümler T ve K'dır.

 (T,K) düğümleri ziyaret edildikten sonra (K, B) düğümü ziyaret edilir.

16



0	1	2	3	4	5	6
A	I	I	Y	L	Z	M

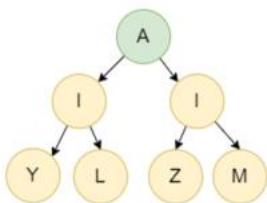
Şekilde bir MinHeap yapısına ve ilgili yapınpın dizi ile temsiline yer verilmiştir. MinHeap yapısınasırasıyla { C } ve { I } karakterileri ekleniyor. Oluşan yeni MinHeap yapısındaki elemanları dizi üzerindeki sıralaması hangi seçenekte doğru olarak verilmiştir?

 A - I - I - Y - L - Z - M - C - I

 C - I - A - I - I - Y - L - Z - M

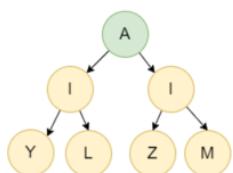
 A - I - I - I - C - Y - L - Z

 A - C - I - I - L - Z - M - Y - I



0	1	2	3	4	5	6
A	I	I	Y	L	Z	M

16



0	1	2	3	4	5	6
A	I	I	Y	L	Z	M



Şekilde bir MinHeap yapısına ve ilgili yapının dizi ile temsiline yer verilmiştir. MinHeap yapısınasırasıyla { I } ve { L } karakterileri ekleniyor. Oluşan yeni MinHeap yapısındaki elemanları dizi üzerindeki sıralaması hangi seçenekte doğru olarak verilmiştir?

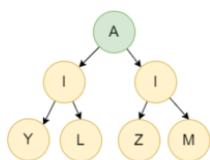
A - I - I - Y - L - Z - M - C - I

C - I - A - I - I - Y - L - Z - M

A - I - I - I - C - Y - L - Z

A - C - I - I - L - Z - M - Y - I

16



0	1	2	3	4	5	6
A	I	I	Y	L	Z	M

Şekilde bir MinHeap yapısına ve ilgili yapının dizi ile temsiline yer verilmiştir. MinHeap yapısınasıyla { C } ve { I } karakterleri ekleniyor. Oluşan yeni MinHeap yapısındaki elemanları dizi üzerindeki sıralaması hangi seçenekte doğru olarak verilmiştir?

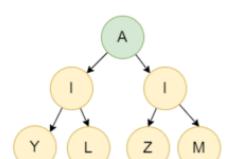
 A - I - I - Y - L - Z - M - C - I

 C - I - A - I - I - Y - L - Z - M

 A - I - I - I - C - Y - L - Z

 A - C - I - I - L - Z - M - Y - I

17



0	1	2	3	4	5	6
A	I	I	Y	L	Z	M

Şekilde bir MinHeap yapısına ve ilgili yapının dizi ile temsiline yer verilmiştir. MinHeap yapısı üzerinden bir eleman silindiğinde kalan elemanların dizi üzerindeki yerleşimi nasıl olur?

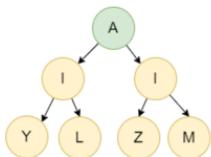
 A - I - I - Y - L - Z

 Z - M - Y - I - L - I

 I - I - Y - L - Z - M

 I - L - I - Y - M - Z

17



0	1	2	3	4	5	6
A	I	I	Y	L	Z	M

Şekilde bir MinHeap yapısına ve ilgili yapının dizi ile temsiline yer verilmiştir. MinHeap yapısı üzerinden bir eleman silindiğinde kalan elemanların dizi üzerindeki yerleşimi nasıl olur?

 A - I - I - Y - L - Z

 Z - M - Y - I - L - I

 I - I - Y - L - Z - M

 I - L - I - Y - M - Z


29



İlk eklenen eleman

Son eklenen eleman

19	23	50	20	71
----	----	----	----	----



Şekilde yığın üzerinde Peek() fonksiyonun çalışması durumunda elde edilecek değer hangi seçenekte verilmiştir?

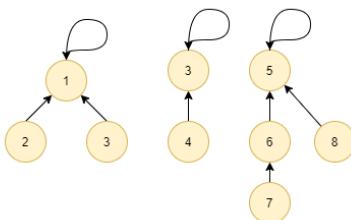
 71

 50

 19

 23

30



Şekilde verilen ayrık setleri inceleyiniz. Buna göre hangi ifade yanlıştır?

Tüm setlerin boyutu (size) eşittir.

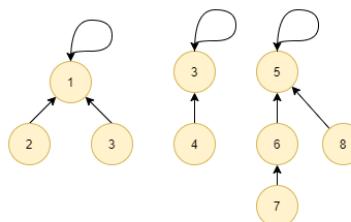


$S_1 = \{1,2,3\}$, $S_2 = \{3,4\}$ ve $S_3 = \{5,6,7,8\}$ şeklinde tanımlanabilir.

Şekilde üç adet ayrık set bulunmaktadır.

Ayrık küme temsilcileri [1], (3) ve (5)'tir.

30



Şekilde verilen ayrık setleri inceleyiniz. Buna göre hangi ifade yanlıştır?

Tüm setlerin boyutu (size) eşittir.

$S_1 = \{1,2,3\}$, $S_2 = \{3,4\}$ ve $S_3 = \{5,6,7,8\}$ şeklinde tanımlanabilir.

Şekilde üç adet ayrık set bulunmaktadır.

Ayrık küme temsilcileri [1], (3) ve (5)'tir.

29



Şekilde yığın üzerinde Peek() fonksiyonun çalışması durumunda elde edilecek değer hangi seçenekte verilmiştir?

 71

 50

 19

 23

28



```
6 references
public SinglyLinkedList(IEnumerable<T> collection)
{
    foreach (var item in collection)
        this.AddLast(item);
}
```

Verilen kod parçasığını inceleyiniz. İlgili koda göre yanlış olan ifade hangisidir?

 Collection parametresi ArrayList olabilir.

 Collection parametresi dizi (Array) olabilir.

 Collection parametresi List<T> olabilir.

 Collection parametresi bir ağaç veri yapısı olabilir.

Algoritma ve Veri Yapıları İleri Seviye

07:05

Verilen kod parçasını inceleyiniz. İlgili koda göre yanlış olan ifade hangisidir?

Collection parametresi ArrayList olabilir.

Collection parametresi dizi (Array) olabilir.

Collection parametresi List<T> olabilir.

Collection parametresi bir ağaç veri yapısı olabilir.

İçerikler Nottam

- 3.Dizi ve Koleksiyonlar (1s 35d 50m / 14/14)
- 4.Bağı Listeler (Linked List) (3s 27d 10m / 23/23)
- 5.Yığın (1s 52m / 6/6)
- 6.Kuyruk (3s 22m / 5/5)
- 7.Ağaç (4s 17d 42m / 24/24)
- 8.Oncelikli Kuyruklar ve Y... (2s 38d 19m / 14/14)
- 9.Ayrık Setler (1s 18d 22m / 13/13)
- 10.Cüngüler (4s 14d 12m / 36/36)
- 11.Cizge Tabanlı Algoritm... (2s 49d 3m / 17/17)

Algoritma ve Veri Yapıları İleri Seviye

06:58

Verilen çizgeyi inceleyiniz. İlgili çizgede referans düğüm yesil renk ile vurgulanmıştır. Bu çizge üzerinde çalışılan bir Kruskal algoritmasına ilişkin ilk ve son ziyaret edilen kenarlar hangi seçenekte doğru olarak verilmiştir?

(B,T) ve (K,A)

(B,T) ve (E,M)

(D,I) ve (E,M)

(D,I) ve (K,A)

İçerikler Nottam

- 3.Dizi ve Koleksiyonlar (1s 35d 50m / 14/14)
- 4.Bağı Listeler (Linked List) (3s 27d 10m / 23/23)
- 5.Yığın (1s 52m / 6/6)
- 6.Kuyruk (3s 22m / 5/5)
- 7.Ağaç (4s 17d 42m / 24/24)
- 8.Oncelikli Kuyruklar ve Y... (2s 38d 19m / 14/14)
- 9.Ayrık Setler (1s 18d 22m / 13/13)
- 10.Cüngüler (4s 14d 12m / 36/36)
- 11.Cizge Tabanlı Algoritm... (2s 49d 3m / 17/17)

Algoritma ve Veri Yapıları İleri Seviye

06:51

27

Referans düğüm

Verilen çizgeyi inceleyiniz. İlgili çizgede referans düğüm yeşil renk ile vurgulanmıştır. Bu çizge üzerinde çalıştırılan bir Kruskal algoritmasına ilişkin ilk ve son ziyaret edilen kenarlar hangi seçenekte doğru olarak verilmiştir?

(B,T) ve (K,A)

(B,T) ve (L,M)

(D,I) ve (E,M)

(D,I) ve (K,A)

< Geri

İleri >

Algoritma ve Veri Yapıları İleri Seviye

Çıkış

İçerikler Notname

- 3.Dizi ve Koleksiyonlar
- 4.Bağı Listeler (Linked List)
- 5.Yığın
- 6.Kuyruk
- 7.Ağaç
- 8.Oncelikli Kuyruklar ve Y...
- 9.Ayrık Setler
- 10.Cüngüler
- 11.Cizge Tabanlı Algoritm...

100%

Algoritma ve Veri Yapıları İleri Seviye

06:08

26

Sekildeki ikili ağaçın PostOrder dolasılması durumunda nasıl bir çıktı elde edilir?

10 - 20 - 30 - 40 - 60 - 70 - 80 - 90

10 - 30 - 20 - 40 - 70 - 90 - 80 - 60

60 - 40 - 80 - 20 - 70 - 90 - 10 - 30

60 - 40 - 20 - 10 - 30 - 80 - 70 - 90

< Geri

İleri >

Algoritma ve Veri Yapıları İleri Seviye

Çıkış

İçerikler Notname

- 3.Dizi ve Koleksiyonlar
- 4.Bağı Listeler (Linked List)
- 5.Yığın
- 6.Kuyruk
- 7.Ağaç
- 8.Oncelikli Kuyruklar ve Y...
- 9.Ayrık Setler
- 10.Cüngüler
- 11.Cizge Tabanlı Algoritm...

100%

Algoritma ve Veri Yapıları İleri Seviye

06:06

25

Şekildeki ikili arama ağacından kök siliniyor. Daha sonra ağaç PostOrder dolaşıyor. Buna göre hangi çıktı doğru dolasımı ifade eder?

70 - 40 - 80 - 20 - 90 - 10 - 30

10 - 20 - 30 - 40 - 70 - 80 - 90

70 - 40 - 20 - 10 - 30 - 80 - 90

10 - 30 - 20 - 40 - 90 - 80 - 70

Geri İleri >

Çıkış

İçerikler Notname

- 3.Dizi ve Koleksiyonlar (1s 35d 50sa (14/14))
- 4.Bağı Listeler (Linked List) (3s 27d 10sa (23/23))
- 5.Yığın (1s 52sa (6/6))
- 6.Kuyruk (1s 17d 22sa (5/5))
- 7.Ağaç (4s 17d 42sa (24/24))
- 8.Oncelikli Kuyruklar ve Y... (2s 38d 19sa (14/14))
- 9.Ayrık Setler (1s 18d 22sa (13/13))
- 10.Cüzgeler (4s 14d 12sa (36/56))
- 11.Çizge Tabanlı Algoritm... (2s 49d 3sa (17/17))

Algoritma ve Veri Yapıları İleri Seviye

05:56

25

Şekildeki ikili arama ağacından kök siliniyor. Daha sonra ağaç PostOrder dolaşıyor. Buna göre hangi çıktı doğru dolasımı ifade eder?

70 - 40 - 80 - 20 - 90 - 10 - 30

10 - 20 - 30 - 40 - 70 - 80 - 90

70 - 40 - 20 - 10 - 30 - 80 - 90

10 - 30 - 20 - 40 - 90 - 80 - 70

Geri İleri >

Çıkış

İçerikler Notname

- 3.Dizi ve Koleksiyonlar (1s 35d 50sa (14/14))
- 4.Bağı Listeler (Linked List) (3s 27d 10sa (23/23))
- 5.Yığın (1s 52sa (6/6))
- 6.Kuyruk (1s 17d 22sa (5/5))
- 7.Ağaç (4s 17d 42sa (24/24))
- 8.Oncelikli Kuyruklar ve Y... (2s 38d 19sa (14/14))
- 9.Ayrık Setler (1s 18d 22sa (13/13))
- 10.Cüzgeler (4s 14d 12sa (36/56))
- 11.Çizge Tabanlı Algoritm... (2s 49d 3sa (17/17))

Algoritma ve Veri Yapıları İleri Seviye

04:17

26

Şekildeki ikili ağac PostOrder dolasılması durumunda nasıl bir çıktı elde edilir?

10 - 20 - 30 - 40 - 60 - 70 - 80 - 90

10 - 30 - 20 - 40 - 70 - 90 - 80 - 60

60 - 40 - 80 - 20 - 70 - 90 - 10 - 30

60 - 40 - 20 - 10 - 30 - 80 - 70 - 90

[Geri](#) [İleri >](#)

Algoritma ve Veri Yapıları İleri Seviye 100%

Çıkış

İçerikler Notname

- 3.Dizi ve Koleksiyonlar 1s 35d 50m (14/14)
- 4.Bağı Listeler (Linked List) 3s 27d 10m (23/23)
- 5.Yığın 1s 52m (6/6)
- 6.Kuyruk 3s 22m (5/5)
- 7.Ağaç 4s 17d 42m (24/24)
- 8.Oncelikli Kuyruklar ve Y... 2s 38d 19m (14/14)
- 9.Ayrık Setler 1s 18d 22m (13/13)
- 10.Cüzgeler 4s 14d 12m (36/36)
- 11.Cizge Tabanlı Algoritm... 2s 49d 3m (17/17)

Algoritma ve Veri Yapıları İleri Seviye

04:00

25

Şekildeki ikili arama ağacında kök siliniyor. Daha sonra ağaç PostOrder dolasılıyor. Buna göre hangi çıktı doğru dolasımı ifade eder?

70 - 40 - 80 - 20 - 90 - 10 - 30

10 - 20 - 30 - 40 - 70 - 80 - 90

70 - 40 - 20 - 10 - 30 - 80 - 90

10 - 30 - 20 - 40 - 90 - 80 - 70

[Geri](#) [İleri >](#)

Algoritma ve Veri Yapıları İleri Seviye 100%

Çıkış

İçerikler Notname

- 3.Dizi ve Koleksiyonlar 1s 35d 50m (14/14)
- 4.Bağı Listeler (Linked List) 3s 27d 10m (23/23)
- 5.Yığın 1s 52m (6/6)
- 6.Kuyruk 3s 22m (5/5)
- 7.Ağaç 4s 17d 42m (24/24)
- 8.Oncelikli Kuyruklar ve Y... 2s 38d 19m (14/14)
- 9.Ayrık Setler 1s 18d 22m (13/13)
- 10.Cüzgeler 4s 14d 12m (36/36)
- 11.Cizge Tabanlı Algoritm... 2s 49d 3m (17/17)

Algoritma ve Veri Yapıları İleri Seviye

03:29

25

Şekildeki ikili arama ağacından kök siliniyor. Daha sonra ağaç PostOrder dolaslıyor. Buna göre hangi çıktı doğru dolasımı ifade eder?

70 - 40 - 80 - 20 - 90 - 10 - 30
 10 - 20 - 30 - 40 - 70 - 80 - 90
 70 - 40 - 20 - 10 - 30 - 80 - 90
 10 - 30 - 20 - 40 - 90 - 80 - 70

< Geri Ileri >

Algoritma ve Veri Yapıları İleri Seviye 100%

Çıkış

İçerikler Notname

- 3.Dizi ve Koleksiyonlar 1s 35d 50sa (14/14)
- 4.Bağı Listeler (Linked List) 3s 27d 10sa (23/23)
- 5.Yığın 1s 52sa (6/6)
- 6.Kuyruk 3s 22d 42sa (5/5)
- 7.Ağaç 4s 17d 42sa (24/24)
- 8.Oncelikli Kuyruklar ve Y... 2s 38d 19sa (14/14)
- 9.Ayrık Setler 1s 18d 22sa (13/13)
- 10.Cüngüler 4s 14d 12sa (36/36)
- 11.Cizge Tabanlı Algoritm... 2s 49d 3sa (17/17)

Algoritma ve Veri Yapıları İleri Seviye

03:26

24

$$T(n) = 6T\left(\frac{n}{3}\right) + n^2 \log n$$

Çalışma zamanı $T(n)$ ile ifade edilen algoritmanın, ana teoreme (master theorem) göre zaman karmaşıklığı hangi seçenekte doğru olarak verilmiştir?

$O(n^2 \log n)$
 $O(n \log n)$
 $O(n)$
 $O(n^2)$

< Geri Ileri >

Algoritma ve Veri Yapıları İleri Seviye 100%

Çıkış

İçerikler Notname

- 3.Dizi ve Koleksiyonlar 1s 35d 50sa (14/14)
- 4.Bağı Listeler (Linked List) 3s 27d 10sa (23/23)
- 5.Yığın 1s 52sa (6/6)
- 6.Kuyruk 3s 22d 42sa (5/5)
- 7.Ağaç 4s 17d 42sa (24/24)
- 8.Oncelikli Kuyruklar ve Y... 2s 38d 19sa (14/14)
- 9.Ayrık Setler 1s 18d 22sa (13/13)
- 10.Cüngüler 4s 14d 12sa (36/36)
- 11.Cizge Tabanlı Algoritm... 2s 49d 3sa (17/17)

Algoritma ve Veri Yapıları İleri Seviye

03:25

Çalışma zamanı $T(n)$ ile ifade edilen algoritmanın, ana teoreme (master theorem) göre zaman karmaşıklığı hangi seçenekte doğru olarak verilmiştir?

$O(n^2 \log n)$

$O(n \log n)$

$O(n)$

$O(n^2)$

« Geri İleri »

Algoritma ve Veri Yapıları İleri Seviye

Çıkış

İçerikler Notname

- 3.Dizi ve Koleksiyonlar (1s 35d 50m / 14/14)
- 4.Bağı Listeler (Linked List) (3s 27d 10m / 23/23)
- 5.Yığın (1s 52m / 6/6)
- 6.Kuyruk (3d 22m / 5/5)
- 7.Ağaç (4s 17d 42m / 24/24)
- 8.Oncelikli Kuyruklar ve Y... (2s 38d 19m / 14/14)
- 9.Ayrık Setler (1s 18d 22m / 13/13)
- 10.Cüzgeler (4s 14d 12m / 36/36)
- 11.Çizge Tabanlı Algoritm... (2s 49d 3m / 17/17)

Type here to search 79% 8:11 PM ENG 6/3/2024

Algoritma ve Veri Yapıları İleri Seviye

03:16

Çalışma zamanı $T(n)$ ile ifade edilen algoritmanın, ana teoreme (master theorem) göre zaman karmaşıklığı hangi seçenekte doğru olarak verilmiştir?

$O(n^2 \log n)$

$O(n \log n)$

$O(n)$

$O(n^2)$

« Geri İleri »

Algoritma ve Veri Yapıları İleri Seviye

Çıkış

İçerikler Notname

- 3.Dizi ve Koleksiyonlar (1s 35d 50m / 14/14)
- 4.Bağı Listeler (Linked List) (3s 27d 10m / 23/23)
- 5.Yığın (1s 52m / 6/6)
- 6.Kuyruk (3d 22m / 5/5)
- 7.Ağaç (4s 17d 42m / 24/24)
- 8.Oncelikli Kuyruklar ve Y... (2s 38d 19m / 14/14)
- 9.Ayrık Setler (1s 18d 22m / 13/13)
- 10.Cüzgeler (4s 14d 12m / 36/36)
- 11.Çizge Tabanlı Algoritm... (2s 49d 3m / 17/17)

Type here to search 79% 8:11 PM ENG 6/3/2024



$$T(n) = 6T\left(\frac{n}{3}\right) + n^2 \log n$$

Çalışma zamanı $T(n)$ ile ifade edilen algoritmanın, ana teoreme (master theorem) göre zaman karmaşıklığı hangi seçenekte doğru olarak verilmiştir?

- $O(n^2 \log n)$
- $O(n \log n)$
- $O(n)$
- $O(n^2)$

Algoritmalar ve Veri Yapıları İleri Seviye

02:32

Algoritmalar ve Veri Yapıları İleri Seviye

İçerikler Nottam 100%

3.Dizi ve Koleksiyonlar 1s 35d 50m (10/14)

4.Bağılı Listeler (Linked List) 3s 27d 10m (23/23)

5.Yığın 1s 52m (6/6)

6.Kuyruk 39d 22m (5/5)

7.Ağaç 4s 17d 42m (24/24)

8.Oncelikli Kuyruklar ve Y... 2s 38d 19m (14/14)

9.Ayrık Setler 1s 18d 22m (13/13)

10.Cüzgeler 4s 14d 12m (36/56)

11.Çerge Tabanlı Algoritm... 2s 49d 3m (17/17)

$f(x) = x^2 + 2x + 1$

Time

Input Size

Yukarıdaki grafiğe göre yanlış olan ifade hangisidir?

- En verimli algoritmaların çalışma süresi sabittir.
- Karesel bir büyümeye orantılı sahip olan algoritma, doğrusal çalışma zamanına sahip olan algoritmdan verimlidir.
- Karmaşıklık arttıkça çalışma zamanı da beraber artmaktadır.
- Sabit bir çalışma zamanına sahip olan algoritmanın büyümeye hızı, giriş boyutundan bağımsızdır.

Geri İleri

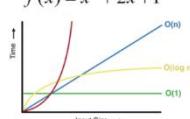
Type here to search

Algoritma ve Veri Yapıları İleri Seviye

02:15

23

$f(x) = x^2 + 2x + 1$



Yukarıdaki grafiğe göre yanlış olan ifade hangisidir?

En verimli algoritmaların çalışma süresi sabittir.

Karesel bir büyümeye orantılı olan algoritmalar, doğrusal çalışma zamanına sahip olan algoritmada verilmemiştir.

Karmazıklık aralığında çalışma zamanı da beraber artmaktadır. 

Sabit bir çalışma zamanına sahip olan algoritmanın büyümeye hızı, giriş boyutundan bağımsızdır.

< Geri | İleri >

Algoritma ve Veri Yapıları İleri Seviye

Çıkış

İçerikler Notname

- 3.Dizi ve Koleksiyonlar (1s 35d 50m / 14/14)
- 4.Bağı Listeler (Linked List) (3s 27d 10m / 23/23)
- 5.Yığın (1s 52m / 6/6)
- 6.Kuyruk (3d 22m / 5/5)
- 7.Ağaç (4s 17d 42m / 24/24)
- 8.Oncelikli Kuyruklar ve Y... (2s 38d 19m / 14/14)
- 9.Ayrık Setler (1s 18d 22m / 13/13)
- 10.Çizgiler (4s 14d 12m / 36/36)
- 11.Çizge Tabanlı Algoritm... (2s 49d 3m / 17/17)

Algoritma ve Veri Yapıları İleri Seviye

02:03

22



Verilen şekli inceleyiniz. İlgili şekilde göre yanlış olan ifade hangisidir?

DeQueue() işlevi sonunda 19 değeri elde edilir.

EnQueue(item) işlevi sonrası eklenen eleman 19'dan sonra gelir.

Silme işleminin maliyeti O(1)'dir.

Ekleme işleminin maliyeti O(n)'dir.

< Geri | İleri >

Algoritma ve Veri Yapıları İleri Seviye

Çıkış

İçerikler Notname

- 3.Dizi ve Koleksiyonlar (1s 35d 50m / 14/14)
- 4.Bağı Listeler (Linked List) (3s 27d 10m / 23/23)
- 5.Yığın (1s 52m / 6/6)
- 6.Kuyruk (3d 22m / 5/5)
- 7.Ağaç (4s 17d 42m / 24/24)
- 8.Oncelikli Kuyruklar ve Y... (2s 38d 19m / 14/14)
- 9.Ayrık Setler (1s 18d 22m / 13/13)
- 10.Çizgiler (4s 14d 12m / 36/36)
- 11.Çizge Tabanlı Algoritm... (2s 49d 3m / 17/17)

Algoritma ve Veri Yapıları İleri Seviye

01:35

22.

Verilen şekli inceleyiniz. İlgili şekilde göre yanlış olan ifade hangisidir?

NULL ← 71 → 20 → 23 → 19 → NULL

Head Tail

ilk eklelen element
Son eklelen element

DeQueue() İşlevi sonunda 19 değeri elde edilir.

EnQueue(item) İşlevi sonrası eklenen eleman 19'dan sonra gelir.

Silme İşlevinin maliyeti O(1)'dir.

Ekleme İşlevinin maliyeti O(n)'dir.

< Geri İleri >

İçerikler Notname

- 3.Dizi ve Koleksiyonlar
- 4.Bağlı Listeler (Linked List)
- 5.Yığın
- 6.Kuyruk
- 7.Ağac
- 8.Oncelikli Kuyruklar ve Y...
- 9.Ayrık Setler
- 10.Cüngüler
- 11.Cizge Tabanlı Algoritm...

Algoritma ve Veri Yapıları İleri Seviye

01:33

21.

Verilen şekti inceleyiniz. İlgili şekilde göre tanımlanmış olan çizge için yanlış olan ifade hangisidir?

A → D: 2.3
A → B: 1.2
D → C: 5.5
E → F: 4.4

Her bir ada bileşen (component) olarak ifade edilir.

İki farklı adadan oluşmaktadır.

Adalar arası bağlantı yoktur.

Güçlü bağlı (strongly connected) bir çizgedir.

< Geri İleri >

İçerikler Notname

- 3.Dizi ve Koleksiyonlar
- 4.Bağlı Listeler (Linked List)
- 5.Yığın
- 6.Kuyruk
- 7.Ağac
- 8.Oncelikli Kuyruklar ve Y...
- 9.Ayrık Setler
- 10.Cüngüler
- 11.Cizge Tabanlı Algoritm...

Algoritma ve Veri Yapıları İleri Seviye

01:26

21.

Verilen şekli inceleyiniz. Ligili şekilde göre tanımlanmış olan çizge için yanlış olan ifade hangisidir?

Her bir ada bileşen (component) olarak ifade edilir.

İki farklı adadan oluşmaktadır.

Adalar arası bağlantı yoktur.

Güçlü bağlı (strongly connected) bir çizgedir.

< Geri İleri >

Çıksız
İçeriğler Nottam
3.Dizi ve Koleksiyonlar
4.Bağı Listeler (Linked List)
5.Yığın
6.Kuyruk
7.Ağaç
8.Oncelikli Kuyruklar ve Y...
9.Ayrık Setler
10.Çizgeler
11.Cizge Tabanlı Algoritm...

Algoritma ve Veri Yapıları İleri Seviye

01:24

21.

Verilen şekli inceleyiniz. Ligili şekilde göre tanımlanmış olan çizge için yanlış olan ifade hangisidir?

Her bir ada bileşen (component) olarak ifade edilir.

İki farklı adadan oluşmaktadır.

Adalar arası bağlantı yoktur.

Güçlü bağlı (strongly connected) bir çizgedir.

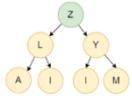
< Geri İleri >

Çıksız
İçeriğler Nottam
3.Dizi ve Koleksiyonlar
4.Bağı Listeler (Linked List)
5.Yığın
6.Kuyruk
7.Ağaç
8.Oncelikli Kuyruklar ve Y...
9.Ayrık Setler
10.Çizgeler
11.Cizge Tabanlı Algoritm...

Algoritma ve Veri Yapıları İleri Seviye

01:06

20.



0	1	2	3	4	5	6
Z	L	Y	A	I	I	M

Şekilde bir Max-Heap yapısına ve ilgili yapının dizi ile temsiline yer verilmiştir. Max-Heap yapısına sırasıyla (C) ve (I) elemanları ekleniyor, buna göre elemanların dizi üzerindeki yerleştirmeleri nasıl olur?

C-I-Z-L-Y-A-I-I-M
 Z-L-Y-I-I-I-M-A-C
 Z-L-Y-A-I-I-M-C-I
 Z-L-Y-A-C-I-I-M

< Geri İleri >

Algoritma ve Veri Yapıları İleri Seviye 100%

Çıkış

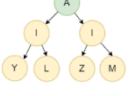
İçerikler Notname

- 3.Dizi ve Koleksiyonlar 1s 35d 50m (14/14)
- 4.Bağı Listeler (Linked List) 3s 27d 10m (23/23)
- 5.Yığın 1s 52m (6/6)
- 6.Kuyruk 3d 22m (5/5)
- 7.Ağaç 4s 17d 42m (24/24)
- 8.Oncelikli Kuyruklar ve Y... 2s 38d 19m (14/14)
- 9.Ayrık Setler 1s 18d 22m (13/13)
- 10.Cüzgeler 4s 14d 12m (36/36)
- 11.Çizge Tabanlı Algoritm... 2s 49d 3m (17/17)

Algoritma ve Veri Yapıları İleri Seviye

00:41

19.



0	1	2	3	4	5	6
A	I	I	Y	L	Z	M

Şekilde bir Min-Heap yapısına ve ilgili yapının dizi ile temsiline yer verilmiştir. Min-Heap yapısında hiç eleman kalmayınca kadar sırasıyla elemanları silmeliyorum. Buna göre silinen elemanların oluşturduğu çıktı hangi seçenekte doğru olarak verilmiştir?

A-I-I-L-M-Y-Z
 A-I-I-Y-L-Z-M
 Z-Y-M-L-I-I-A
 M-Z-L-Y-I-I-A

< Geri İleri >

Algoritma ve Veri Yapıları İleri Seviye 100%

Çıkış

İçerikler Notname

- 3.Dizi ve Koleksiyonlar 1s 35d 50m (14/14)
- 4.Bağı Listeler (Linked List) 3s 27d 10m (23/23)
- 5.Yığın 1s 52m (6/6)
- 6.Kuyruk 3d 22m (5/5)
- 7.Ağaç 4s 17d 42m (24/24)
- 8.Oncelikli Kuyruklar ve Y... 2s 38d 19m (14/14)
- 9.Ayrık Setler 1s 18d 22m (13/13)
- 10.Cüzgeler 4s 14d 12m (36/36)
- 11.Çizge Tabanlı Algoritm... 2s 49d 3m (17/17)

Algoritma ve Veri Yapıları İleri Seviye

00:38

19

Şekilde bir Mini-Heap yapısına ve ilgili yapının dizi ile temsiline yer verilmiştir. Mini-Heap yapısında hiç eleman kalmayınca kadar sırasıyla elemanlar silinir. Buna göre silinen elemanların oluşturduğu çıktı hangi seçenekte doğru olarak verilmiştir?

A-I-I-L-M-Y-Z

A-I-I-Y-L-Z-M

Z-Y-M-L-I-I-A

M-Z-L-Y-I-I-A

Geri İleri >

İçerikler Notname

3.Dizi ve Koleksiyonlar 1s 35d 50m (14/14)

4.Bağı Listeler (Linked List) 3s 27d 10m (23/23)

5.Yığın 1s 52m (6/6)

6.Kuyruk 3d 22m (5/5)

7.Ağaç 4s 17d 42m (24/24)

8.Oncelikli Kuyruklar ve Y... 2s 38d 19m (14/14)

9.Ayrık Setler 1s 18d 22m (13/13)

10.Cüzgeler 4s 14d 12m (36/36)

11.Çizge Tabanlı Algoritm... 2s 49d 3m (17/17)

Algoritma ve Veri Yapıları İleri Seviye

00:22

18

Şekilde bir Max-Heap yapısına ve ilgili yapının dizi ile temsiline yer verilmiştir. Max-Heap yapısından bir eleman silinmesi durumunda; kalan elemanları dizi üzerindeki yerleştirmeleri nasıl olur?

I-I-A-M-L-Y

L-Y-A-I-I-M

Z-L-Y-A-I-I

Y-L-M-A-I-I

Geri İleri >

İçerikler Notname

3.Dizi ve Koleksiyonlar 1s 35d 50m (14/14)

4.Bağı Listeler (Linked List) 3s 27d 10m (23/23)

5.Yığın 1s 52m (6/6)

6.Kuyruk 3d 22m (5/5)

7.Ağaç 4s 17d 42m (24/24)

8.Oncelikli Kuyruklar ve Y... 2s 38d 19m (14/14)

9.Ayrık Setler 1s 18d 22m (13/13)

10.Cüzgeler 4s 14d 12m (36/36)

11.Çizge Tabanlı Algoritm... 2s 49d 3m (17/17)

Algoritma ve Veri Yapıları İleri Seviye

00:05

Verilen şekli inceleyiniz. İlgili şekilde göre tanımlanmış olan çizge için yanlış olan ifade hangisidir?

Her bir ada bileşen (component) olarak ifade edilir.

İki farklı adadan oluşmaktadır.

Adalar arası bağlantı yoktur.

Güçlü bağlı (strongly connected) bir çizgedir.

< Geri İleri >

İçerikler Notname

- 3.Dizi ve Koleksiyonlar (14/14)
- 4.Bağı Listeler (Linked List) (23/23)
- 5.Yığın (6/6)
- 6.Kuyruk (5/5)
- 7.Ağaç (24/24)
- 8.Oncelikli Kuyruklar ve Y... (14/14)
- 9.Ayrık Setler (13/13)
- 10.Çizgeler (36/36)
- 11.Çizge Tabanlı Algoritm... (17/17)

Type here to search 8:14 PM 6/3/2024

Algoritma ve Veri Yapıları İleri Seviye

00:02

Verilen şekli inceleyiniz. İlgili şekilde göre yanlış olan ifade hangisidir?

DeQueue() işlevi sonunda 19 değeri elde edilir.

EnQueue(item) işlevi sonrası eklenen eleman 19'dan sonra gelir.

Silme işleminin maliyeti O(1)'dir.

Ekleme işleminin maliyeti O(n)'dir.

< Geri İleri >

İçerikler Notname

- 3.Dizi ve Koleksiyonlar (14/14)
- 4.Bağı Listeler (Linked List) (23/23)
- 5.Yığın (6/6)
- 6.Kuyruk (5/5)
- 7.Ağaç (24/24)
- 8.Oncelikli Kuyruklar ve Y... (14/14)
- 9.Ayrık Setler (13/13)
- 10.Çizgeler (36/36)
- 11.Çizge Tabanlı Algoritm... (17/17)

Type here to search 8:15 PM 6/3/2024

Elbette, veri yapıları sınavı için çeşitli konular ve örnek sorular hazırlayabilirim. Veri yapıları, bilgisayar bilimlerinin temel konularından biridir ve genellikle şu başlıklarını kapsar:

1. Temel Veri Yapıları

- Diziler
- Bağlı Listeler (Singly Linked List, Doubly Linked List, Circular Linked List)
- Yığınlar (Stacks)
- Kuyruklar (Queues) ve Çift uçlu kuyruklar (Deques)

2. Ağaçlar

- İkili Ağaçlar (Binary Trees)
- İkili Arama Ağaçları (Binary Search Trees - BST)
- Denge Ağaçları (AVL Trees, Red-Black Trees)
- Heap Ağaçları (Binary Heaps, Min-Heaps, Max-Heaps)
- B-Ağaçları (B-Trees)

3. Grafik

- Temel Grafik Teorisi (Vertices, Edges, Paths, Cycles)
- Grafik Temsilleri (Adjacency Matrix, Adjacency List)
- Derinlemesine Arama (Depth-First Search - DFS)
- Genişlemesine Arama (Breadth-First Search - BFS)
- Minimum Örtücü Ağaçlar (Minimum Spanning Trees - Kruskal, Prim)
- En Kısa Yol Algoritmaları (Dijkstra, Bellman-Ford)

4. Hashing

- Hash Tabloları
- Çarpışma Çözme Teknikleri (Chaining, Open Addressing)

5. Sıralama ve Arama Algoritmaları

- Sıralama (Bubble Sort, Selection Sort, Insertion Sort, Merge Sort, Quick Sort, Heap Sort)
- Arama (Binary Search, Linear Search)

Örnek sınav soruları:

1. Temel Veri Yapıları Sorusu

Soru: Bağlı bir listede bir düğümü silmek için kullanılan genel algoritmayı yazın ve zaman karmaşıklığını açıklayın.

Cevap:

[markdown](#)

[Copy code](#)

1.

2.

3.

2. Ağaçlar Sorusu

Soru: Bir ikili arama ağacının (BST) denge faktörünü açıklayın ve AVL ağacının nasıl çalıştığını özetleyin.

Cevap:

[markdown](#)

[Copy code](#)



3. Grafik Sorusu

Soru: Depth-First Search (DFS) algoritmasını açıklayın ve bir örnek üzerinde çalıştırın.

Cevap:

[markdown](#)

[Copy code](#)

1.

2.

3.

4.

DFS Sırası: A, B, E, F, C, D, G

[bash](#)

[Copy code](#)

hash

in

hash

in

hash

hash

Bu soruların her biri, farklı veri yapıları konularını kapsamaktadır. Öğrencilerin bu soruları cevaplayarak konuların detaylarını anlamaları ve algoritmaları pratik etmeleri beklenir.

Kuyruk veri yapısı, doğrusal olmayan veri yapıları için yardımcı bir veri yapısı olarak kullanılamaz. false true

ChatGPT

"Kuyruk veri yapısı, doğrusal olmayan veri yapıları için yardımcı bir veri yapısı olarak kullanılamaz." ifadesi yanlıştır (false).

Kuyruk veri yapısı, doğrusal olmayan veri yapıları için sıkça kullanılan yardımcı bir veri yapısıdır. Örneğin:

1. **Ağaç Taraması:** Genişlik öncelikli arama (BFS) algoritmasında, ağaç veya grafiği katman katman taramak için kuyruk veri yapısı kullanılır. BFS, bir başlangıç düğümünden başlayarak, tüm komşu düğümleri bir kuyruktaki sıraya koyar ve sırayla işler.
2. **Grafik Algoritmaları:** Yine BFS algoritmasında, grafiklerin bağlantılı bileşenlerini keşfetmek, en kısa yol problemlerini çözmek veya minimum örtücü ağaçları bulmak için kuyruklar kullanılır.
3. **Yazıcı İş Kuyruğu:** Bir yazıcının işlerini sıraya koymak için kuyruk kullanılır, bu da sistemdeki diğer doğrusal olmayan yapıların verimli bir şekilde çalışmasını sağlar.

Bu nedenle, kuyruklar doğrusal olmayan veri yapıları için de önemli bir yardımcı araçtır.

Özetle, "Kuyruk veri yapısı, doğrusal olmayan veri yapıları için yardımcı bir veri yapısı olarak kullanılamaz." ifadesi **yandalıktır**.

Diziler düzenli ya da düzensiz olarak tanımlanabilir. false true

ChatGPT

"Diziler düzenli ya da düzensiz olarak tanımlanabilir." ifadesi **doğrudur** (true).

Diziler, elemanlarının belirli bir sıraya göre düzenlenip düzenlenmediğine göre iki farklı kategoriye ayrılabilir:

1. Düzenli Diziler (Sorted Arrays):

- Elemanları belirli bir düzene göre sıralanmış dizilerdir.
- Örneğin, artan ya da azalan sıraya göre sıralanmış olabilirler.
- Sıralı dizilerde arama, ekleme ve silme gibi işlemler genellikle daha verimli olabilir.

2. Düzensiz Diziler (Unsorted Arrays):

- Elemanları belirli bir düzene göre sıralanmamış dizilerdir.
- Elemanlar herhangi bir sırada olabilir.
- Bu tür dizilerde işlemler genellikle daha az verimli olabilir çünkü belirli bir düzen olmadığı için arama, ekleme ve silme işlemleri daha fazla zaman alabilir.

Dolayısıyla, diziler hem düzenli hem de düzensiz olarak tanımlanabilirler. Bu, dizi elemanlarının sıralı olup olmamasına bağlıdır.

3



Tek yönlü bağlı listede liste sonundan eleman silmenin maliyeti $O(1)$ dir.

true

false

< Geri

İleri >

ChatGPT

Tek yönlü bağlı listede sondan eleman silmenin maliyeti $O(1)$ değildir. Bu ifade yanlıştır (false).

Açıklama: Tek yönlü bağlı listede bir elemanı sondan silmek için, sondaki düğümün öncesindeki düğümün (yani sondan bir önceki düğümün) bulunması gereklidir. Bu, genellikle listenin başından başlayarak düğümleri tek tek dolaşıp sondan bir önceki düğüme ulaşmayı gerektirir. Listenin her bir düğümünü kontrol ederek sondan bir önceki düğüme ulaşmak $O(n)$ zaman alır, çünkü n elemanı içeren bir listede n-1 düğümü kontrol etmeniz gerekebilir.

Bu nedenle, tek yönlü bağlı bir listede sondan eleman silmenin zaman karmaşıklığı $O(1)$ değil, $O(n)$ olur.

Aşağıdakilerden hangisi Bağlı Listelerin avantajlarından biri değildir? Liste başına ekleme işlevi sabit zamanda gerçekleştirilir. Liste elemanları üzerinde dolaşmanın maliyeti $O(n)$ 'dir. Listenin başından silme işlevi sabit zamanda gerçekleştirilir. Dinamik bellek yönetimi yapılmasına olanak verir.

ChatGPT

Bağlı listelerin avantajlarından biri olmayan ifade şudur:

Liste elemanları üzerinde dolaşmanın maliyeti $O(n)$ 'dir.

Bu ifade, bağlı listelerin bir avantajı değil, bir sınırlamasıdır. Bağlı listelerin avantajları genellikle şunlardır:

- Liste başına ekleme işlevi sabit zamanda gerçekleştirilir ($O(1)$):** Bağlı listelerde, bir elemanı listenin başına eklemek sabit zamanda yapılabilir.
- Liste başından silme işlevi sabit zamanda gerçekleştirilir ($O(1)$):** Listenin başındaki bir elemanı silmek de sabit zamanda yapılabilir.
- Dinamik bellek yönetimi yapılmasına olanak verir:** Bağlı listeler, bellek yönetimi açısından esneklik sağlar ve gerektiğinde belleği dinamik olarak ayırip serbest bırakabilir.

Ancak, liste elemanları üzerinde dolaşmanın maliyeti $O(n)O(n)$ olması, bağlı listelerin bir avantajı değil, dezavantajıdır. Bağlı listelerde elemanlar üzerinde dolaşmak için listenin başından sonuna kadar sırasıyla ilerlemek gereklidir, bu da zaman karmaşıklığı açısından $O(n)O(n)$ maliyetlidir.

Diziler sabit boyutlu olarak tanımlanır. false true

ChatGPT

"Diziler sabit boyutlu olarak tanımlanır." ifadesi **doğrudur** (true).

Açıklama: Diziler, genellikle tanımlandıkları anda belirlenen ve daha sonra değiştirilemeyen sabit bir boyuta sahiptir. Bu, dizilerin bellek alanı yönetiminde belirli bir alanın önceden ayrılmaması anlamına gelir. Sabit boyutlu dizilerde (statik diziler), bir kez oluşturulduktan sonra dizinin boyutu değiştirilmez.

Örneğin, C ve Java gibi birçok programlama dilinde bir dizi tanımlandığında, bu dizinin boyutu sabit olur:

```
c
Copy code
int 10
```

Bu dizinin boyutu 10 olarak tanımlanmıştır ve daha sonra bu boyut değiştirilemez.

Dinamik diziler veya dinamik olarak tahsis edilen bellek yapıları farklıdır ve bunların boyutu programın çalışması sırasında değiştirilebilir. Ancak, genel olarak ve temel tanımlamaya göre, diziler sabit boyutlu olarak tanımlanır.

Dizi boyutunun dinamik olarak yönetilmesi isteniyor ise ArrayList ya da Generic List kullanılabilir. true false

ChatGPT

"Dizi boyutunun dinamik olarak yönetilmesi isteniyor ise ArrayList ya da Generic List kullanılabilir." ifadesi **doğrudur** (true).

Açıklama: Birçok modern programlama dilinde, standart diziler sabit boyutludur ve tanımlandıktan sonra boyutları değiştirilemez. Ancak, dinamik olarak büyüyebilen ve küçülebilen veri yapıları vardır. Örneğin:

- **Java'da ArrayList:** Java'da ArrayList, dinamik olarak boyutunu ayarlayabilen bir dizidir. Eleman ekledikçe veya çıkardıkça, ArrayList otomatik olarak yeniden boyutlandırılır.

```
java
Copy code
new ArrayList[1]
0
```

- **C#'ta List<T>:** C# dilinde Generic List (List<T>), benzer şekilde dinamik boyutlandırma sağlar.

csharp

[Copy code](#)



Bu veri yapıları, altında yatan diziyi gerektiğinde yeniden boyutlandırarak çalışır, böylece kullanıcıların boyutla ilgili endişelenmesine gerek kalmaz. Bu nedenle, dinamik boyut yönetimi gereken durumlarda ArrayList (Java) veya Generic List (C#) gibi yapıların kullanılması doğrudur.

Tek yönlü bağlı listede liste başından eleman silmenin maliyeti doğrusaldır. false true

ChatGPT

"Tek yönlü bağlı listede liste başından eleman silmenin maliyeti doğrusaldır." ifadesi **yanlıştır** (false).

Açıklama: Tek yönlü bağlı listede liste başından eleman silmenin maliyeti sabit zamandır ($O(1)$). Liste başındaki elemanı silmek için yalnızca baş işaretçisini bir sonraki düğüme ayarlamak yeterlidir. Bu işlem, liste ne kadar büyük olursa olsun sabit zamanda gerçekleşir. Dolayısıyla, maliyeti doğrusal ($O(n)$) değil, sabit ($O(1)$) zaman karmaşıklığına sahiptir.

markdown

[Copy code](#)

Tek yönlü bağlı listede arama yapmanın maliyeti $O(1)$ 'dir. false true

ChatGPT

"Tek yönlü bağlı listede arama yapmanın maliyeti $O(1)$ 'dir." ifadesi **yanlıştır** (false).

Açıklama: Tek yönlü bağlı listede bir elemanı aramak için genellikle listenin başından başlayarak her bir düğümü sırasıyla kontrol etmek gerekir. Bu işlem, aranan elemanın konumuna bağlı olarak en kötü durumda listenin sonuna kadar devam edebilir. Dolayısıyla, arama işleminin zaman karmaşıklığı $O(n)$ olur, burada n listenin eleman sayısını temsil eder.

markdown

[Copy code](#)



Bu nedenle, tek yönlü bağlı listede arama yapmanın maliyeti $O(1)$ değil, $O(n)$ zaman karmaşıklığına sahiptir.

Tek yönlü bağlı listede arama yapmanın maliyeti $O(n)$ 'dir. false true

ChatGPT

"Tek yönlü bağlı listede arama yapmanın maliyeti $O(n)$ 'dir." ifadesi **doğrudur** (true).

Açıklama: Tek yönlü bağlı listede bir elemanı aramak için genellikle listenin başından başlayarak her bir düğümü sırasıyla kontrol etmek gereklidir. Bu işlem, aranan elemanın konumuna bağlı olarak en kötü durumda listenin sonuna kadar devam edebilir. Dolayısıyla, arama işleminin zaman karmaşıklığı $O(n)$ olur, burada n listenin eleman sayısını temsil eder.

[markdown](#)

[Copy code](#)



Bu nedenle, tek yönlü bağlı listede arama yapmanın maliyeti $O(n)$ zaman karmaşıklığına sahiptir.