

VERİ YAPILARILARI VE ALGORİTMALAR

Array, Dynamic Array and Collections

Dizi

Array

Single variable

1

Array: Indexes

0

1

2

3

4

Values

1

3

8

23

99

Dizi

Array

- En temel veri yapılarından biridir.
- Dizi elemanlarına erişmek üzere genellikle sıfır-tabanlı indisleme (**zero-based indexing**) kullanılır.
- Tek boyutlu (**single-dimension**) ya da çok-boyutlu (**multi-dimension**) olabilir.
- Düzenli (**regular**) ya da düzensiz (**jagged**) olarak tanımlanabilir.

Dizi

Array

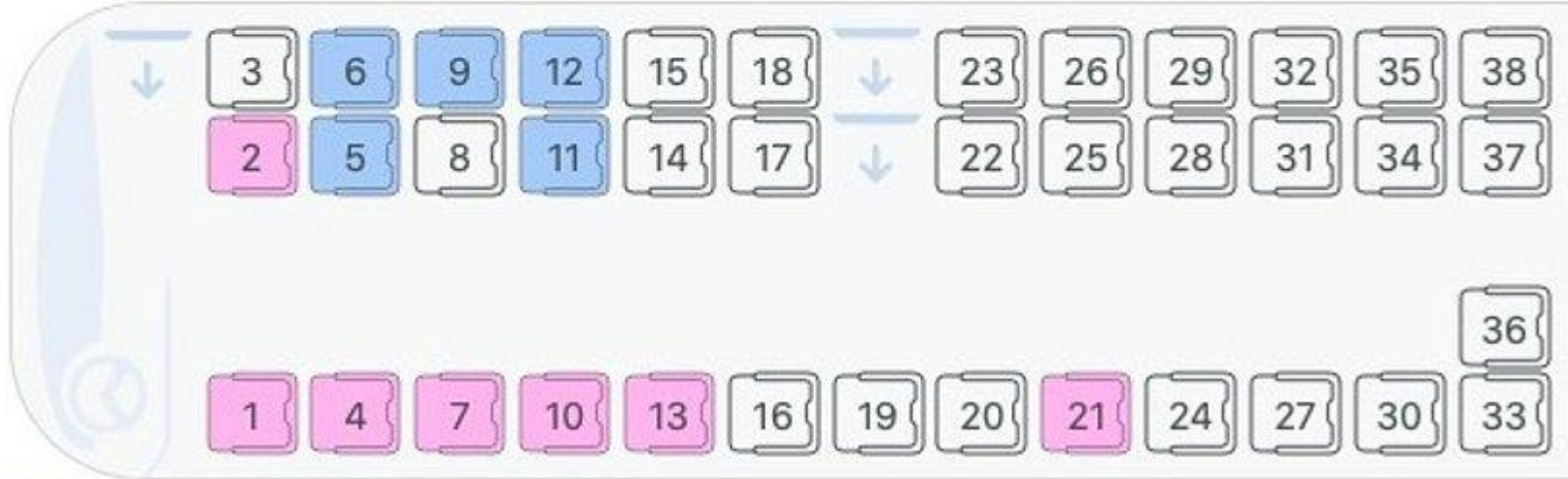
- En büyük dezavantajı **sabit boyutlu** olmasıdır.
- Dizinin sabit boyutlu olmasından dolayı ekleme ve silme gibi işlemlerin maliyeti artar.
- En büyük avantajı ise bellek gözlerine doğrudan erişimin olmasıdır.

2+1
Wi-Fi

12:00 → 19:00
7sa

63,00TL

Erkek - Dolu Kadın - Dolu Boş Koltuk



Sepetim

✓ Tümüü Seç

Mağaza : CBTXGlobal

Mağaza Puanı : %100 ?



Veikk S640 6 x 4" 8192 Levels
5080 LPI Grafik Tablet + Kalem
RENK: Siyah



Sil



Favorilere Ekle

Ürün fiyatı değişti.

Ücretsiz Kargo

- 3 Adet +

749,70 TL

Mağaza : muhtesempazar

Mağaza Puanı : %100 ?



P47 Wireless SD Kart Destekli
Bluetooth 5.0 Kulak Üstü Kulaklık
RENK: Beyaz



Sil



Favorilere Ekle

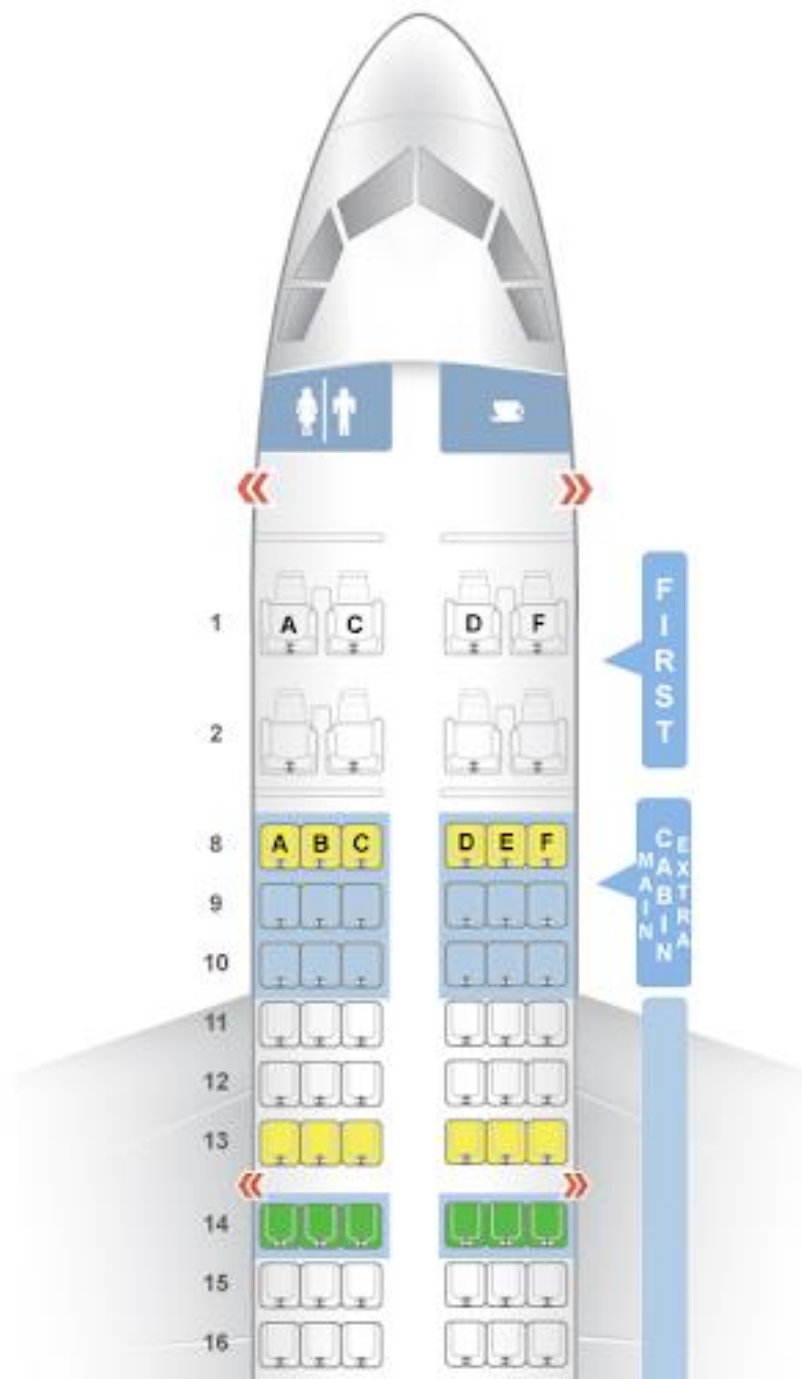
Ürün bilgileri ve fiyatı değişti.

Ücretsiz Kargo

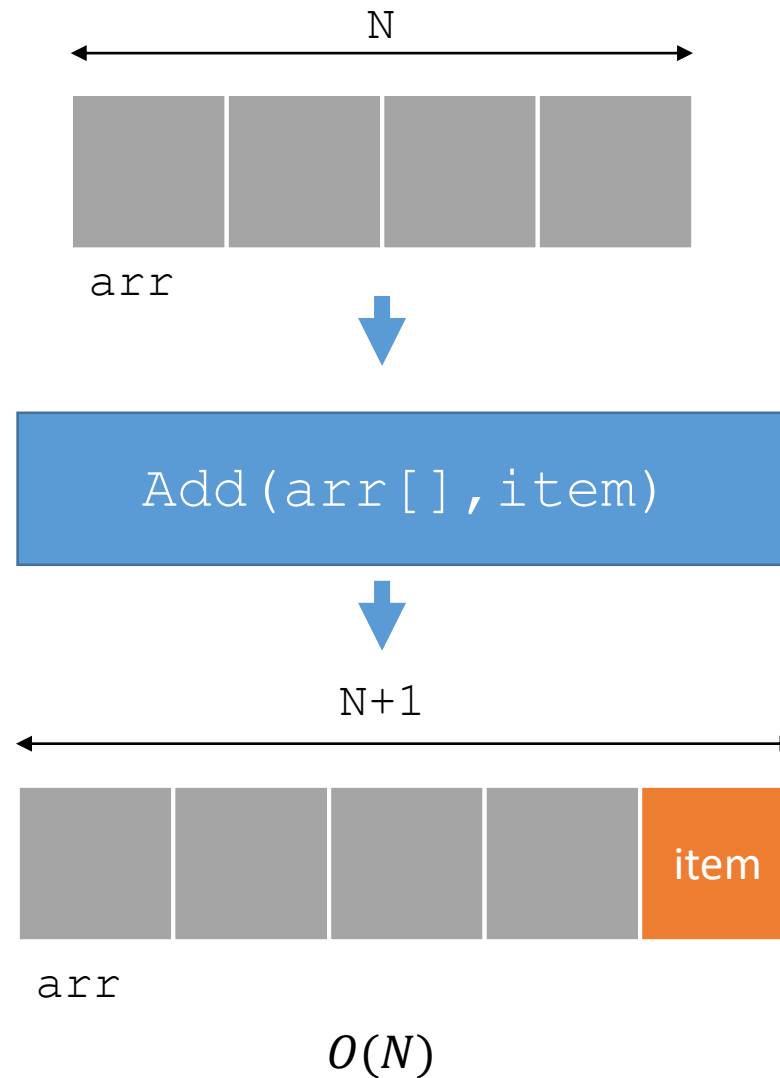
- 1 Adet +

42,55 TL

Results		Messages
	Id	Title
1	1	Akademik Bilişim
2	4	Matlab Dersleri Başlıyor ...
3	7	İmge Bölütleme
4	9	Mobil Web Sayfası Yayında ...
5	10	Programlama Nedir?
6	11	Programlama Dünyasında Sizi Neler Bekliyor?
7	12	CSS Nedir?
8	13	CSS Kullanımı
9	14	Biçimlendime
10	15	Listeler



Add(arr[],item)



Add(arr[], item)

input

arr[] : Array
item : To be added item

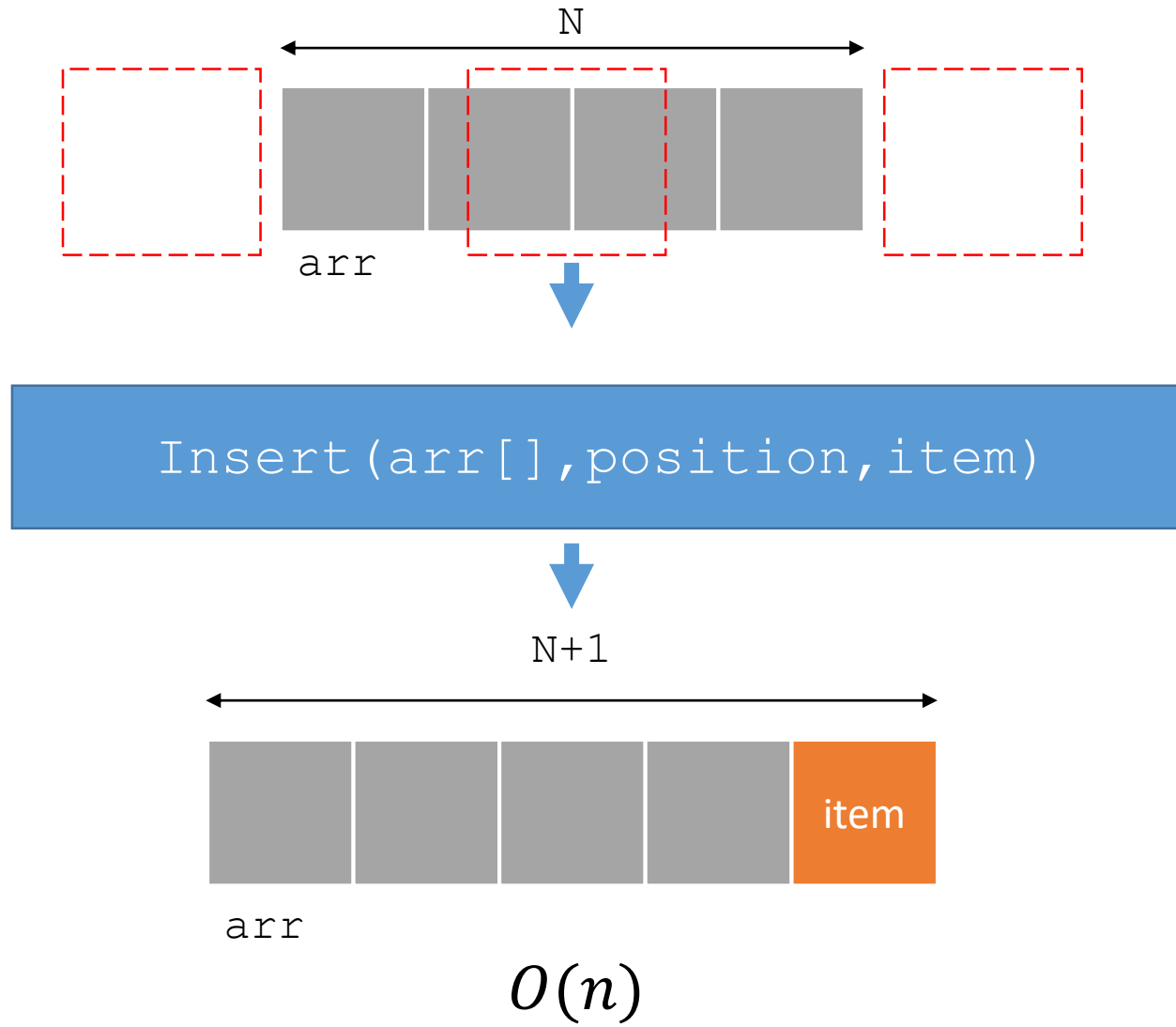
output

arr[] : New Array including new item

```
temp = [];  
for i = 1 to N  
    temp[i] = arr[i];  
end  
temp[array.Length] = item;  
arr = temp;
```

$O(n)$

Insert(arr[], position, item)



Insert(arr[], position, item)

input

arr[] : Array
position : Position on the Array.
item : To be added item

output

arr[] : New Array including new item

```
temp = [];  
for i = 1 to N  
    if (i < position)  
        temp[i] = arr[i]  
    else  
        temp[i+1] = arr[i];  
    endif  
endfor  
temp[position] = item;  
arr = temp;
```

$$O(n)$$

RemoveAt(arr[], position)

input

arr[] : Array
position : Position on the Array.

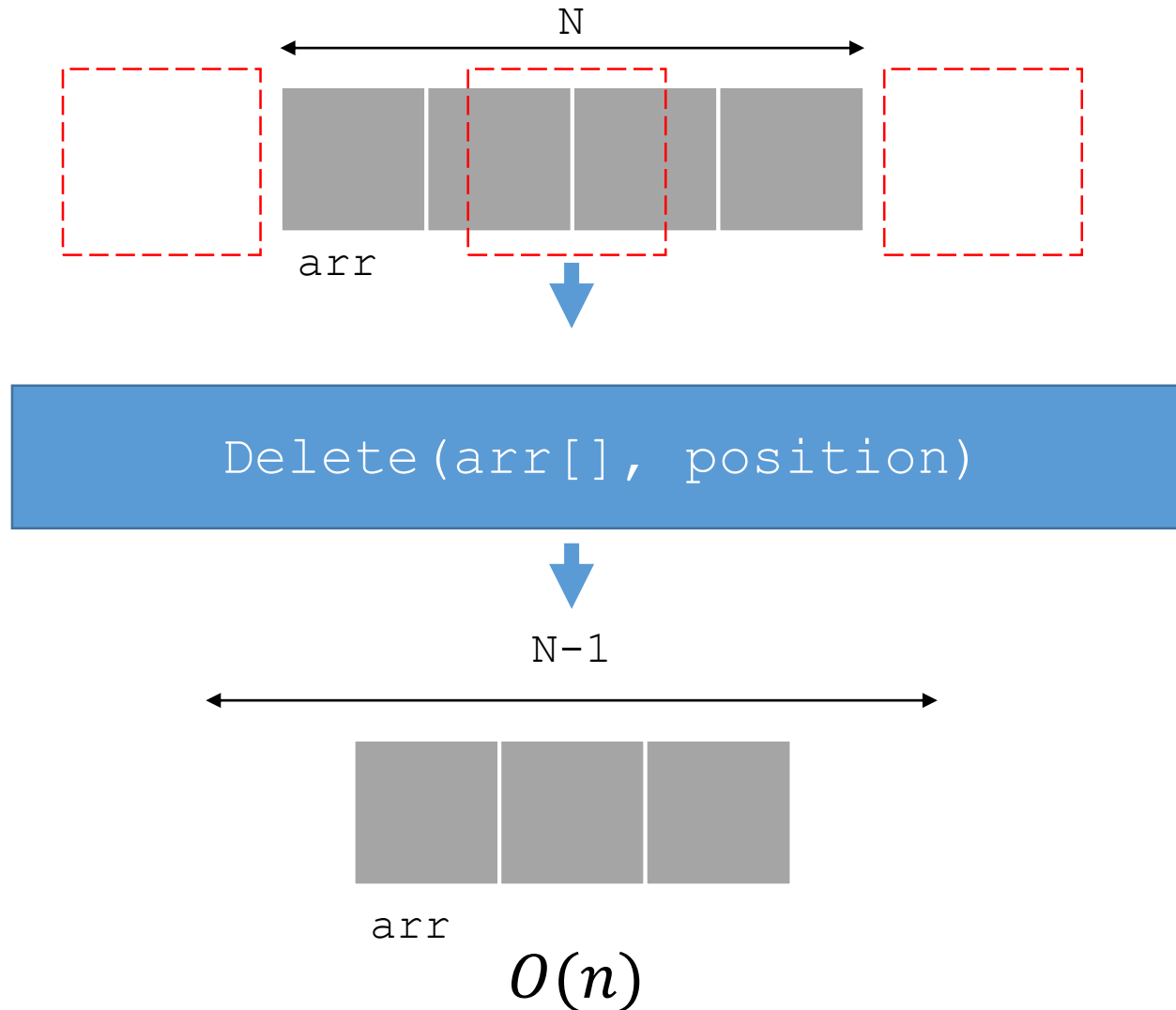
output

arr[] : New Array

```
temp = [];  
for i = 1 to N  
    if (i < position)  
        temp[i] = arr[i]  
    endif  
    if (i > position)  
        temp[i-1] = arr[i];  
    endif  
endfor  
arr = temp;
```

$O(n)$

RemoveAt(arr[], position)



IndexOf(arr[], key)

input

arr[] : Array
key : The key element.

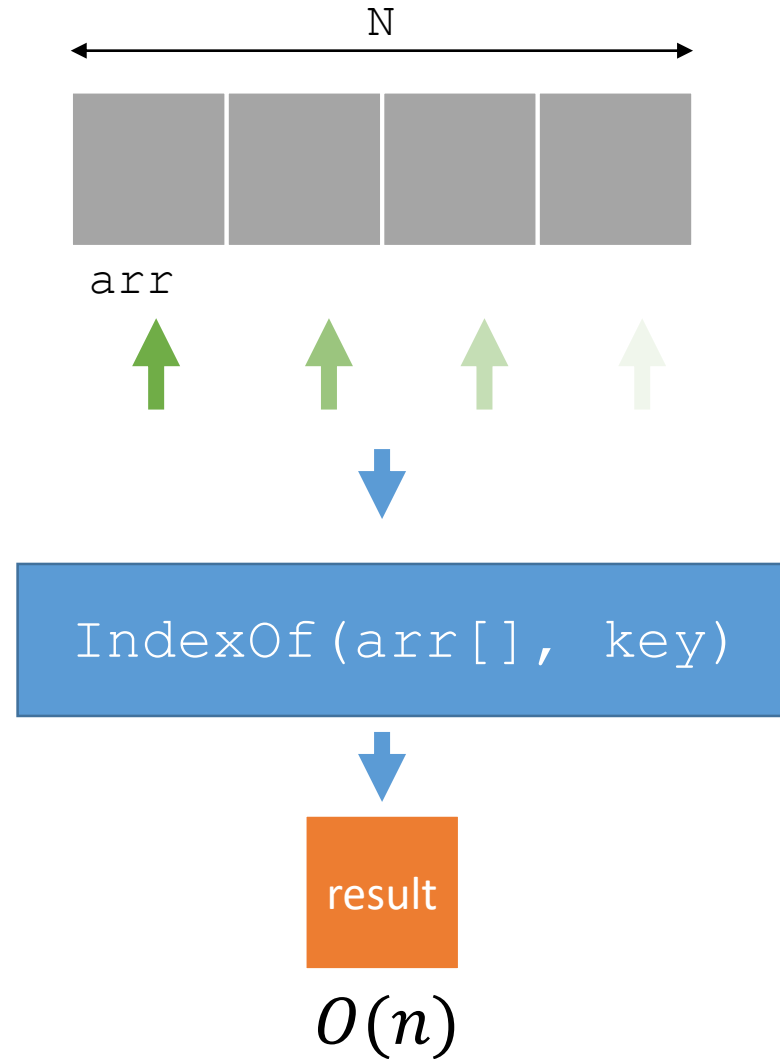
output

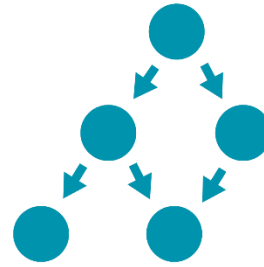
result : The position of the key element or -1.

```
temp = [];  
result = -1;  
for i = 1 to N  
    if (key==arr[i])  
        result = i;  
        return result;  
    endif  
endfor  
return result;
```

$$O(n)$$

IndexOf(arr[], key)





Veri Yapıları ve Algoritmalar

ZAFER CÖMERT

Öğretim Üyesi