

Veri Yapıları ve Algoritmalar

2006-2007 2.dönem

Öğretim Elemanları:

Dr. A. Şima Etaner-Uyar

Dr. Gülşen Cebiroğlu-Eryiğit

Dersle ilgili bilgiler

- Ders Kitabı

[Data Structures and Algorithms in Java](#), 4th Ed., Michael T. Goodrich, Roberto Tamassia, Wiley, 2006.

- Web sayfası

http://www3.itu.edu.tr/~etaner/courses/BT_VeriYap_07/

- Sunular

<http://www3.itu.edu.tr/~gulsenc/dersler/btveri/btveri.html>

Notlandırma

Vize sınavı	1 adet	%30
Ödev	3 adet	%30
Final sınavı	1 adet	%40

Ders Planı

1	12.1.2007	Diziler
2	19.1.2007	Bağlantılı listeler
3	26.1.2007	Uygulama
4	2.2.2007	Bağlantılı listeler Rekürsiyon
5	9.2.2007	Yığın ve kuyruk yapıları
6	16.2.2007	Uygulama
7	23.2.2007	Arasnav
8	2.3.2007	Listeler ve iteratörler
9	9.3.2007	Ağaçlar
10	16.3.2007	Uygulama
11	23.3.2007	Öncelik kuyukları Haritalar ve sözlükler
12	30.3.2007	Arama ağaçları
13	6.4.2007	Uygulama

Ders Akışı

- Dizilerle kayıt saklama
- Kayıt ekleme
- Kayıt silme
- Sıralama (eklemeli)
- İki boyutlu diziler

Kayıt

Ör: Bir oyunda oyuncuların aldığı en yüksek skorların tutulacağı bir veri yapısı

Oyuncu adı	Skor
------------	------

Ayşe	1500
------	------

Dizi

- Kayıtları, alınan skorlara göre sıralı olarak tutan veri yapısı
- $Dizi_boyutu = Dizide\ tutulmak\ istenen\ kayıtların\ sayısı$
- $Dizi_boyutu = 5$

0		Ayşe	1500
1		Mert	1250
2		Ali	1000
3		Ooo	500
4			

Sıralamaya girmek

- Oyunu ilk defa oynayan sıralamaya muhakkak girer.
- Oyun ilk oynayan dizi_boyutu kadar kişi sıralamaya girer.
- $(dizi_boyutu+1)$. Kişi için farklı durumlar oluşabilir:

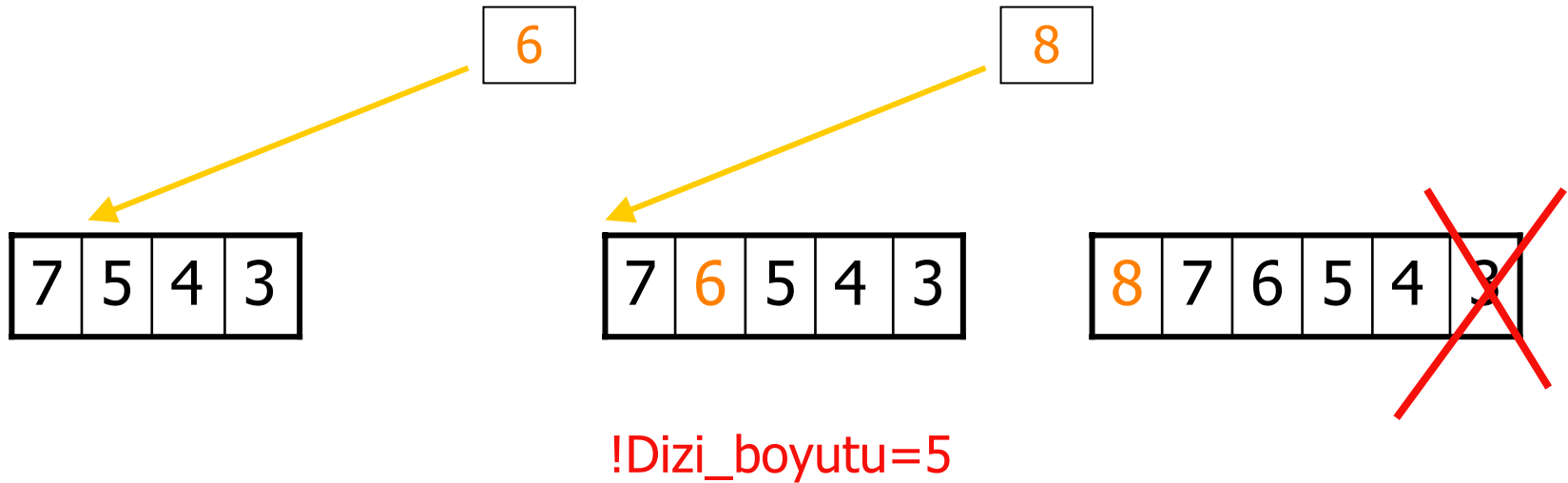
Sıralamaya girmek

- $(dizi_boyutu+1)$. Kişi için farklı durumlar oluşabilir:
 - Eğer skoru dizideki en düşük skordan küçükse diziyeye girmez.
 - Eğer skoru dizideki en yüksek skordan büyükse dizinin en başına geçer.
 - Eğer oyuncu sıralamaya girebilecek bir skor elde ettiyse, diziyeye girer, en düşük skorlu oyuncu diziden çıkarılır.

Dizi nasıl sıralı tutulur?

- Kayıtlar diziye sıralı olarak eklenir.
- Sıralamaya girmeye hak kazanan oyuncuya ait kayıt doğru yer bulunarak yerleştirilir.
- Bu yerleştirme sonucunda dizi boyutundan daha fazla kayıt oluşuyorsa en düşük skorlu olan çıkarılır.

Dizi nasıl sıralı tutulur?



Kayıt (<http://ww0.java4.datastructures.net/source/>)

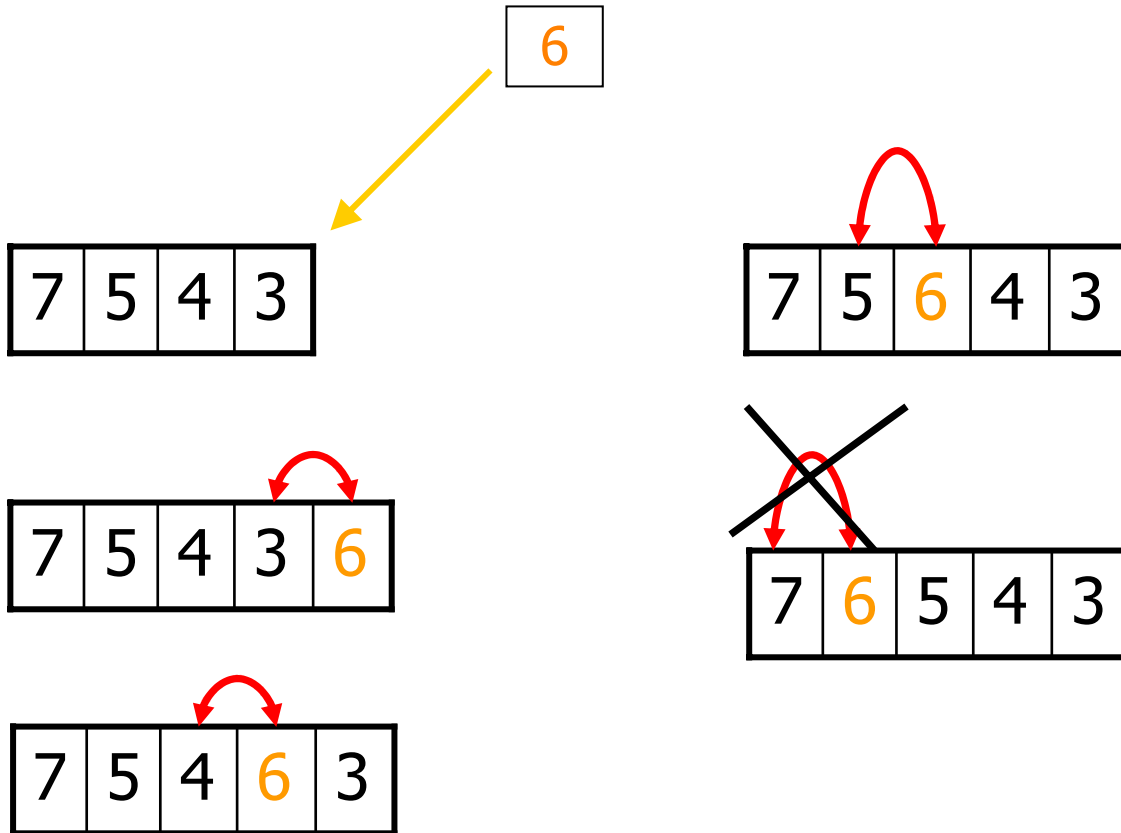
```
public class GameEntry {
    protected String name; // name of the person earning this score
    protected int score; // the score value
    /** Constructor to create a game entry */
    public GameEntry(String n, int s) {
        name = n;
        score = s;
    }
    /** Retrieves the name field */
    public String getName() { return name; }
    /** Retrieves the score field */
    public int getScore() { return score; }
    /** Returns a string representation of this entry */
    public String toString() { return "(" + name + ", " + score + ")"; }
}
```

```

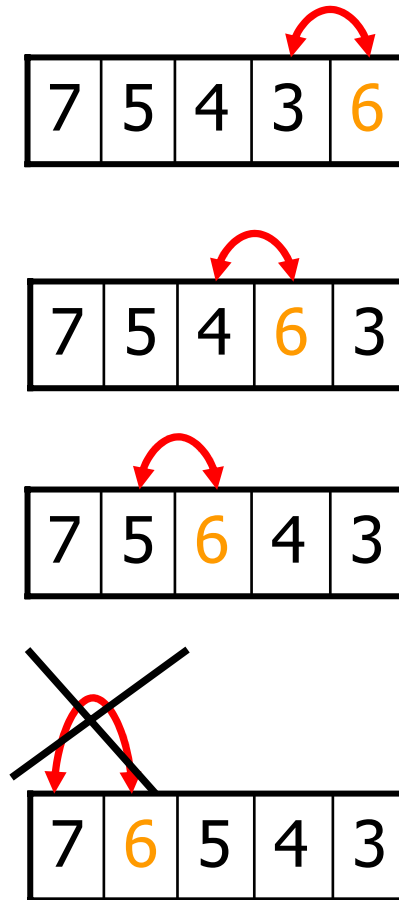
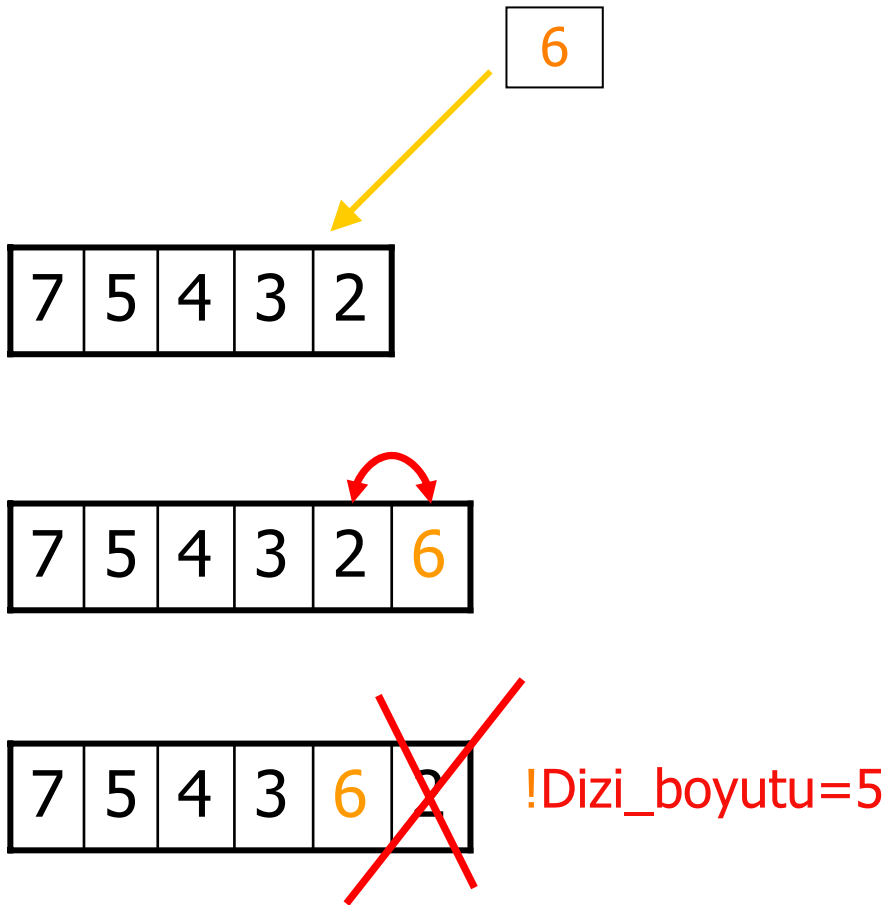
/** Kayıtları tutmak için oluşturulan veri yapısı */
public class Scores {
    public static final int maxEntries = 10; // number of high scores we keep
    protected int numEntries; // number of actual entries
    protected GameEntry[] entries; // array of game entries (names & scores)
    /** Default constructor */
    public Scores() {
        entries = new GameEntry[maxEntries];
        numEntries = 0;
    }
    /** Returns a string representation of the high scores list */
    public String toString() {
        String s = "[";
        for (int i = 0; i < numEntries; i++) {
            if (i > 0) s += ", "; // separate entries by commas
            s += entries[i];
        }
        return s + "]";
    } // ... methods for updating the set of high scores go here ...
}

```

Sıralı kayıt ekleme



Sıralı kayıt ekleme



```

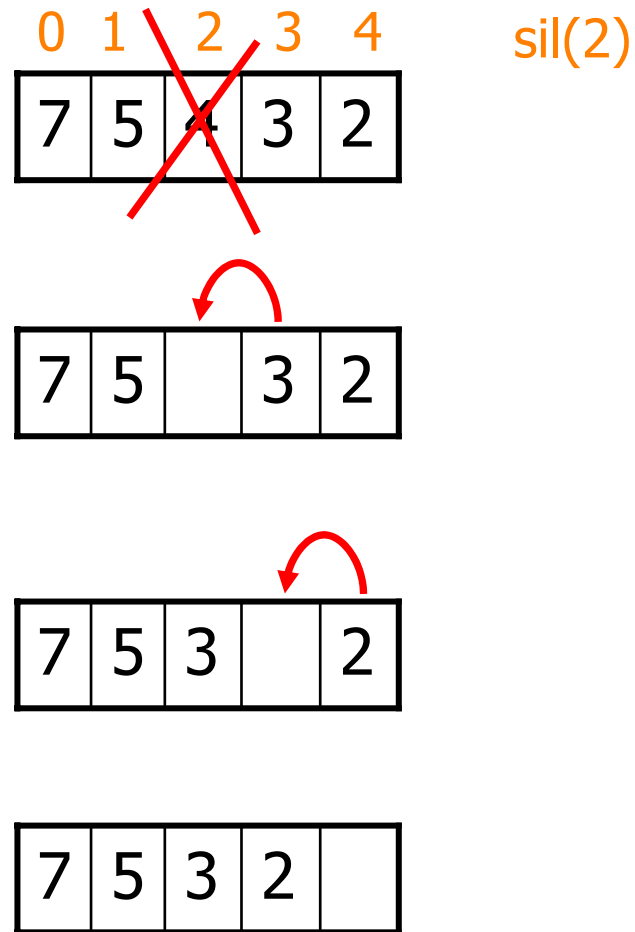
/** Attempt to add a new score to the collection (if it is high enough) */
public void add(GameEntry e) {
    int newScore = e.getScore();
    // is the new entry e really a high score?
    if (numEntries == maxEntries) { // the array is full
        if (newScore <= entries[numEntries-1].getScore())
            return; // the new entry, e, is not a high score in this case
    }
    else // the array is not full
        numEntries++;
    // Locate the place that the new (high score) entry e belongs
    int i = numEntries-1;
    for ( ; (i >= 1) && (newScore > entries[i-1].getScore()); i--)
        entries[i] = entries[i - 1]; // move entry i one to the right
    entries[i] = e; // add the new score to entries
}

```

Kayıt silme

- İstenilen indisteki kayıdı silmek ve silinen kayıdın indisini geri döndürmek.
- Eğer verilen indis doğru aralıkta değilse hata döndürülmelidir.
indis < 0 veya indis > dizi_boyutu
- Dizi boş kalan yeri doldurmak üzere güncellenmelidir.

Kayıt silme



```

/** Remove and return the high score at index i */
public GameEntry remove(int i) throws IndexOutOfBoundsException
{
    if ((i < 0) || (i >= numEntries))
        throw new IndexOutOfBoundsException("Invalid index: " + i);
    GameEntry temp = entries[i]; // temporarily save the object to be removed
    for (int j = i; j < numEntries - 1; j++) // count up from i (not down)
        entries[j] = entries[j+1]; // move one cell to the left
    entries[numEntries - 1 ] = null; // null out the old last score
    numEntries--;
    return temp; // return the removed object
}

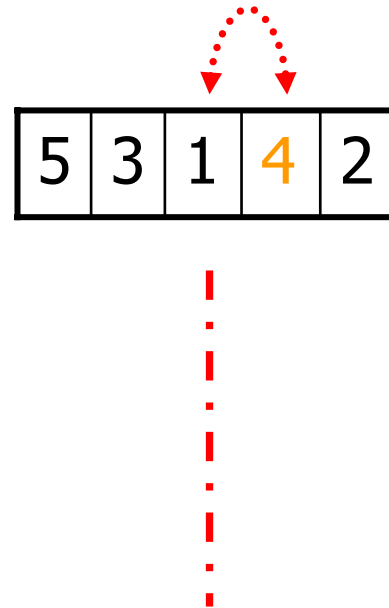
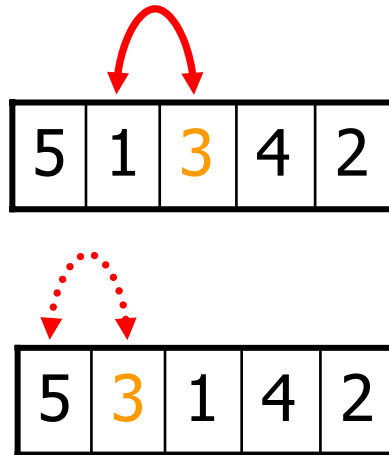
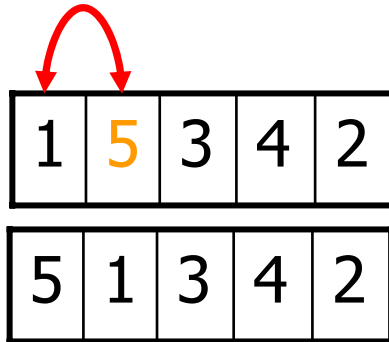
```

Varolan bir diziyi sıralama

1	5	3	4	2
---	---	---	---	---

5	4	3	2	1
---	---	---	---	---

Eklemeli sıralama (insertion sort)



Eklemeli sıralama

```
for i ← 1 to n-1 do
  cur ← A[i]
  j ← i-1
  while j ≥ 0 and a[j] < cur do
    A[j+1] ← A[j]
    j ← j-1
  A[j+1] ← cur
```

```

/** Insertion sort of an array of characters into non-decreasing order */
public static void insertionSort(char[] a) {
    int n = a.length;
    for (int i = 1; i < n; i++) { // index from the second character in a
        char cur = a[i]; // the current character to be inserted
        int j = i - 1; // start comparing with cell left of i
        while ((j >= 0) && (a[j] < cur)) //while a[j] is out of order with cur
            a[j + 1] = a[j--]; // move a[j] right and decrement j
        a[j + 1]=cur; // this is the proper place for cur
    }
}

```

Java.util.Arrays

- equals(A,B)
- fill(A,x)
- sort(A)
- toString(A)

İki boyutlu diziler

```
int [][] Y = new int[4][5];
```

	0	1	2	3	4
0	4	7			
1					
2					
3					

$$Y[i][i+1]=Y[i][i]+3$$

$$i=0$$

$$Y[0][1]=Y[0][0]+3$$
$$= 4+3=7$$

$$i=Y.length=4$$

$$j=Y[2].length=5$$