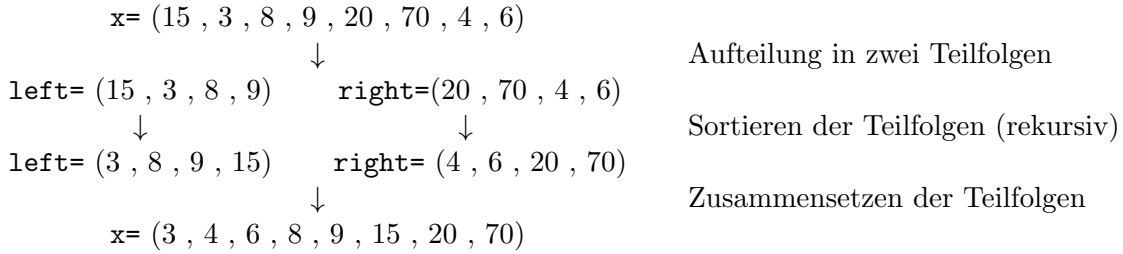
Prof. Dr. W. Rülling
Fakultät Informatik

9.11.20

Algorithmen und Datenstrukturen
Übung 5 (AIN2)

1) Im Vorlesungsteil "Effiziente Algorithmen und Datenstrukturen" wurde das *Mergesort*-Sortierverfahren vorgestellt. Es funktioniert so, dass man ein großes Feld mit Zahlen zunächst in zwei kleinere Felder etwa halber Länge aufgeteilt, die kleineren Teilfelder dann (rekursiv) sortiert und schließlich die beiden sortierten Felder zu einem großen Feld zusammenmischt.



Implementieren Sie dieses Verfahren durch eine **rekursive** Methode
`void mergesort(Integer[] x)`
in einer Klasse `Sortierung` und testen Sie Ihr Programm mit Hilfe des folgenden Hauptprogramms.

```
public class Sortierungsbeispiel
{
 public static void main(String[] args)
   {Integer[] a= {15, 3, 8, 9, 20, 70, 4, 6};
    Sortierung.mergesort(a);
    System.out.print("a: ");
    for (int i=0; i<a.length; ++i)
      System.out.print(a[i] + " ");
    System.out.println();
   }
}
```

Hinweis: Halten Sie sich bitte an den oben skizzierten Ablauf des Algorithmus und orientieren Sie sich nicht an der im Skript angegebenen C++ Implementierung.

2) Bei der Implementierung des *Mergesort*-Verfahrens fällt auf, dass der Algorithmus prinzipiell auch für andere Klassen als `Integer` verwendet werden kann. Man muss nur voraussetzen können, dass man jeweils zwei Objekte miteinander vergleichen kann.

   Zum Vergleichen von Objekten gibt es in Java das Interface `Comparable`. Es beinhaltet eine Methode `int compareTo(Object o)`, mit der geprüft werden kann, ob das aktuelle Objekt größer, kleiner oder gleich dem Objekt `o` ist. Als Ergebnis erhält man einen positiven Wert für größer, einen negativen für kleiner und 0 für die Gleichheit.

```
public interface Comparable
{
  public int compareTo(Object o);
}
```

   Viele in Java verfügbaren Klassen, wie z.B. `Integer`, `Double`, `String`, u.s.w. implementieren dieses Interface `Comparable`.

   Schreiben Sie Ihre Methode `mergesort` so um, dass sie als Parameter statt `Integer[] x` den Parameter `Comparable[] x` verwendet. Dann kann dann die Methode für alle Datentypen verwendet werden, die das Interface `Comparable` implementieren.

   Testen Sie Ihre Implementierung für Felder von Strings.

   Hinweise: Das Interface `Comparable` ist bereits in Java verfügbar. Sie sollten es als nicht selber programmieren!
   Eventuelle Warnungen, dass `Comparable` nicht typsicher ist, können Sie vorerst ignorieren.

3) Erweitern Sie die Klasse `Bruch` aus Übung 2, so dass sie ebenfalls das Interface `Comparable` implementiert und lassen Sie mit `mergesort` ein Array mit Brüchen sortieren.