



Technische  
Universität  
Braunschweig



# Programmieren 1 – Übung #8

Arne Schmidt

# Heute



Rekursion

Entrekursivierung

Live Coding

# Teilbarkeitsregel 17

Eine Zahl  $n$  ist ganzzahlig durch 17 teilbar, wenn die Summe von zwei Mal die Einerstelle plus drei Mal die restlichen Ziffern durch 17 teilbar ist.

## Beispiel:

$3158 \rightarrow 945 + 16 = 961 \rightarrow 288 + 2 = 290 \rightarrow 87 = 24 + 14 = 38 \rightarrow 9 + 16 = 25 \rightarrow 6 + 10 = 16$

3158 ist nicht durch 17 teilbar!

## Beispiel:

$1479 \rightarrow 441 + 18 = 459 \rightarrow 135 + 18 = 153 \rightarrow 45 + 6 = 51$   
 $\rightarrow 15 + 2 = 17$

1479 ist durch 17 teilbar!

Implementiere einen rekursiven Algorithmus, welcher testet, ob eine Zahl durch 17 teilbar ist.

Entrekursiviere diesen anschließend!

# Mergesort

Algorithmus (grob):

1. Sortiere linke Hälfte
2. Sortiere rechte Hälfte
3. Füge beide Hälften zusammen (merge)

| $A =$    | 4 | 2 | 6 | 3 | 1 | 5 | 7 |
|----------|---|---|---|---|---|---|---|
| 1. $A =$ |   |   |   |   |   |   |   |
| 2. $A =$ |   |   |   |   |   |   |   |
| 3. $A =$ |   |   |   |   |   |   |   |
| 4. $A =$ |   |   |   |   |   |   |   |
| 5. $A =$ |   |   |   |   |   |   |   |
| 6. $A =$ |   |   |   |   |   |   |   |

# Mergesort

Algorithmus (grob):

1. Sortiere linke Hälfte
2. Sortiere rechte Hälfte
3. Füge beide Hälften zusammen (merge)

| $A =$    | 4 | 2 | 6 | 3 | 1 | 5 | 7 |
|----------|---|---|---|---|---|---|---|
| 1. $A =$ | 2 | 4 |   |   |   |   |   |
| 2. $A =$ |   |   |   |   |   |   |   |
| 3. $A =$ |   |   |   |   |   |   |   |
| 4. $A =$ |   |   |   |   |   |   |   |
| 5. $A =$ |   |   |   |   |   |   |   |
| 6. $A =$ |   |   |   |   |   |   |   |

# Mergesort

Algorithmus (grob):

1. Sortiere linke Hälfte
2. Sortiere rechte Hälfte
3. Füge beide Hälften zusammen (merge)

| $A =$    | 4 | 2 | 6 | 3 | 1 | 5 | 7 |
|----------|---|---|---|---|---|---|---|
| 1. $A =$ | 2 | 4 |   |   |   |   |   |
| 2. $A =$ |   |   | 3 | 6 |   |   |   |
| 3. $A =$ |   |   |   |   |   |   |   |
| 4. $A =$ |   |   |   |   |   |   |   |
| 5. $A =$ |   |   |   |   |   |   |   |
| 6. $A =$ |   |   |   |   |   |   |   |

# Mergesort

Algorithmus (grob):

1. Sortiere linke Hälfte
2. Sortiere rechte Hälfte
3. Füge beide Hälften zusammen (merge)

| $A =$    | 4 | 2 | 6 | 3 | 1 | 5 | 7 |
|----------|---|---|---|---|---|---|---|
| 1. $A =$ | 2 | 4 |   |   |   |   |   |
| 2. $A =$ |   |   | 3 | 6 |   |   |   |
| 3. $A =$ | 2 | 3 | 4 | 6 |   |   |   |
| 4. $A =$ |   |   |   |   |   |   |   |
| 5. $A =$ |   |   |   |   |   |   |   |
| 6. $A =$ |   |   |   |   |   |   |   |

# Mergesort

Algorithmus (grob):

1. Sortiere linke Hälfte
2. Sortiere rechte Hälfte
3. Füge beide Hälften zusammen (merge)

| $A =$    | 4 | 2 | 6 | 3 | 1 | 5 | 7 |
|----------|---|---|---|---|---|---|---|
| 1. $A =$ | 2 | 4 |   |   |   |   |   |
| 2. $A =$ |   |   | 3 | 6 |   |   |   |
| 3. $A =$ | 2 | 3 | 4 | 6 |   |   |   |
| 4. $A =$ |   |   |   |   | 1 | 5 |   |
| 5. $A =$ |   |   |   |   |   |   |   |
| 6. $A =$ |   |   |   |   |   |   |   |



# Mergesort

Algorithmus (grob):

1. Sortiere linke Hälfte
2. Sortiere rechte Hälfte
3. Füge beide Hälften zusammen (merge)

| $A =$    | 4 | 2 | 6 | 3 | 1 | 5 | 7 |
|----------|---|---|---|---|---|---|---|
| 1. $A =$ | 2 | 4 |   |   |   |   |   |
| 2. $A =$ |   |   | 3 | 6 |   |   |   |
| 3. $A =$ | 2 | 3 | 4 | 6 |   |   |   |
| 4. $A =$ |   |   |   |   | 1 | 5 |   |
| 5. $A =$ |   |   |   |   | 1 | 5 | 7 |
| 6. $A =$ |   |   |   |   |   |   |   |

# Mergesort

Algorithmus (grob):

1. Sortiere linke Hälfte
2. Sortiere rechte Hälfte
3. Füge beide Hälften zusammen (merge)

Implementiere Mergesort.

| $A =$    | 4 | 2 | 6 | 3 | 1 | 5 | 7 |
|----------|---|---|---|---|---|---|---|
| 1. $A =$ | 2 | 4 |   |   |   |   |   |
| 2. $A =$ |   |   | 3 | 6 |   |   |   |
| 3. $A =$ | 2 | 3 | 4 | 6 |   |   |   |
| 4. $A =$ |   |   |   |   | 1 | 5 |   |
| 5. $A =$ |   |   |   |   | 1 | 5 | 7 |
| 6. $A =$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 |