

# Datenstrukturen & Algorithmen

## Übungsblatt 06 | Tutorium 2

Danje Petersen 379748

13.06.2018

### Aufgabe H16

quicksort() wird rekursiv aufgerufen.

#### **quicksort(0,5):**

Anfang: [4,1,3,2,5,9]

Ende: [2,1,3,4,5,9]

#### **quicksort(0,2):**

Anfang: [2,1,3]

Ende: [1,2,3]

#### **quicksort(4,5):**

Anfang: [5,9]

Ende: [5,9]

### Aufgabe H17

Man braucht ein Hilfsarray mit k Einträgen. Man zählt wie oft jeder Eintrag vorkommt und trägt diese Anzahl pro Wert in das Hilfsarray ein.

Danach addiert man die Zahlen so auf, dass man an der Stelle für jeden Wert die letzte zukünftige Adresse dieses Wertes hat.

Man geht dann das ursprüngliche Array von hinten nach vorne durch und setzt den Wert den man vorfindet in ein neues Array der selben Länge an die letzte Adresse, welche im Hilfsarray gespeichert ist und zählt dann eben diesen Eintrag um eins herunter.

## Aufgabe H18

### a) Heapsort:

[10, 9, 3, 4, 8, 6, 5, 2]  
[2, 9, 3, 4, 8, 6, 5, 10]  
[9, 2, 6, 4, 8, 3, 5, 10]  
[5, 2, 6, 4, 8, 3, 9, 10]  
[8, 5, 6, 4, 2, 3, 9, 10]  
[3, 5, 6, 4, 2, 8, 9, 10]  
[6, 5, 3, 4, 2, 8, 9, 10]  
[2, 5, 3, 4, 6, 8, 9, 10]  
[5, 2, 3, 4, 6, 8, 9, 10]  
[4, 2, 3, 5, 6, 8, 9, 10]  
[4, 2, 3, 5, 6, 8, 9, 10]  
[3, 2, 4, 5, 6, 8, 9, 10]  
[2, 3, 4, 5, 6, 8, 9, 10]  
[2, 3, 4, 5, 6, 8, 9, 10]

### b) Mergesort:

[3, 9, 6, 4, 8, 10, 5, 2]  
[3, 9, 6, 4] [8, 10, 5, 2]  
[3, 9] [6, 4] [8, 10] [5, 2]  
[3] [9] [6] [4] [8] [10] [5] [2]  
[3, 9] [4, 6] [8, 10] [2, 5]  
[3, 4, 6, 9] [2, 5, 8, 10]  
[2, 3, 4, 5, 6, 8, 9, 10]