

Auftrag zu Mergesort

Olivier Warin

28. Dezember 2022

1. Sortieren Sie die folgende Liste sowohl mit Mergesort als auch mit Bubblesort.

15	16	19	11	13	20	17	12
----	----	----	----	----	----	----	----

Abbildung 1: Zu sortierende Liste

Zählen Sie bei beiden Verfahren, wie oft Sie zwei Zahlen miteinander vergleichen.

Hinweis: Schneiden Sie die Zahlen aus, um so einfacher Umordnen zu können. Sie finden daher hier noch 4 Kopien der Liste.

15	16	19	11	13	20	17	12
15	16	19	11	13	20	17	12
15	16	19	11	13	20	17	12
15	16	19	11	13	20	17	12
15	16	19	11	13	20	17	12

Abbildung 2: Kopien der Liste zum ausschneiden

2. Implementieren Sie Mergesort mit Hilfe der Vorlage auf moodle.

Vergleichen Sie die Laufzeiten von Mergesort und Bubblesort für zufällig generierte Arrays von den Längen 10, 1000 und 100000. Sie finden alle entsprechenden Funktionen in der Vorlage.

```
import numpy as np
import time

def merge(array1, array2):
    merged_list=[];
    [i,j]=[0,0];
    while i<len(array1) or j<len(array2):
        if i >=len(array1):
            merged_list.append(array2[j])
            j += 1
        elif j>=len(array2) or array1[i] < array2[j]:
            merged_list.append(array1[i])
            i += 1
        elif array2[j] <= array1[i]:
            merged_list.append(array2[j])
            j += 1
    return merged_list

def first_half(array):
    return array[:len(array)//2]

def second_half(array):
    #Implementieren Sie hier eine Funktion, die
    # - passend zu first_half - die zweite
    #Hälfte eines arrays zurück gibt.
    return array

def merge_sort(array):
    #Implementieren Sie hier Mergesort rekursiv
    return array

def bubble_sort(array):
    n = len(array)
    swapped = False
    for i in range(n):
        for j in range(n-i-1):
            if array[j] > array[j+1]:
                swapped = True
                array[j], array[j+1] = array[j+1], array[j]
        if not swapped:
            return array
    return array

def time_sorting_algorithm(sorting_algorithm,array):
    start_time=time.time()
    sorting_algorithm(array)
    end_time=time.time()
    return end_time-start_time

def create_random_array(length):
    return np.random.rand(length)

# Führen Sie hier ihre Befehle aus
```

Quellcode 1: Vorlage für Mergesort in Python, wie sie auf moodle zu finden ist.