

Auftrag zu Mergesort

Olivier Warin

28. Dezember 2022

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



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.



Abbildung 2: Zu sortierende Liste (Kopie 1)



Abbildung 3: Zu sortierende Liste (Kopie 2)



Abbildung 4: Zu sortierende Liste (Kopie 3)



Abbildung 5: Zu sortierende Liste (Kopie 4)

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.

```

1  import numpy as np
2  import time
3
4  def merge(array1, array2):
5      merged_list=[];
6      [i,j]=[0,0];
7      while i<len(array1) or j<len(array2):
8          if i >=len(array1):
9              merged_list.append(array2[j])
10             j += 1
11             elif j>=len(array2) or array1[i] < array2[j]:
12                 merged_list.append(array1[i])
13                 i += 1
14                 elif array2[j] <= array1[i]:
15                     merged_list.append(array2[j])
16                     j += 1
17             return merged_list
18
19 def first_half(array):
20     return array[:len(array)//2]
21
22 def second_half(array):
23     #Implementieren Sie hier eine Funktion, die
24     # - passend zu first_half - die zweite
25     #Hälfte eines arrays zurück gibt.
26     return array
27
28 def merge_sort(array):
29     #Implementieren Sie hier Mergesort rekursiv
30     return array
31
32 def bubble_sort(array):
33     n = len(array)
34     swapped = False
35     for i in range(n):
36         for j in range(n-i-1):
37             if array[j] > array[j+1]:
38                 swapped = True
39                 array[j], array[j+1] = array[j+1], array[j]
40         if not swapped:
41             return array
42     return array
43
44 def time_sorting_algorithm(sorting_algorithm,array):
45     start_time=time.time()
46     sorting_algorithm(array)
47     end_time=time.time()
48     return end_time-start_time
49
50 def create_random_array(length):
51     return np.random.rand(length)
52
53 #Führen Sie hier ihre Befehle aus

```

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