

8.4.2025 12:39:58

merge_sort.py

Page 1/2

```

1
2 # HSLU / ICS/AI ML : Modul ADS : Algorithmen & Datenstrukturen
3 # Path : uebung08/al/aufgabe01
4 # Version: Tue Apr 8 12:39:58 CEST 2025
5
6 import random
7 import sys
8 from time import time
9
10
11 def merge_sort(s):
12     """
13     Sorts a list with the merge-sort algorithm.
14
15     Precondition: Length must be 2^x.
16     s: Sequence to be sorted.
17     Returns the sorted sequence.
18     """
19     n = len(s)
20     if n > 1:
21         s1, s2 = partition(s, n//2)
22
23         # TODO: Implement here...
24
25     return s
26
27
28 def partition(s, n):
29
30     # TODO: Implement here...
31
32     return None
33
34
35 def merge(a, b):
36     n = len(a) * 2
37     s = [None] * n
38     ai = 0 # First element in 'sequence' A
39     bi = 0 # First element in 'sequence' B
40     si = 0 # Last element in 'sequence' S
41
42     # TODO: Implement here...
43
44     return s
45
46
47 def verify(originalData, sortedData):
48     correctSorted = originalData.copy()
49     correctSorted.sort()
50     for i in range(len(originalData)):
51         if(correctSorted[i] != sortedData[i]):
52             print("ERROR: wrong sorted!")
53             print("Original : ", originalData)
54             print("Sorted : ", sortedData)
55             print(f"index[{i}]: should be: {correctSorted[i]}, but is: {sortedData[i]}")
56             sys.exit(1)
57
58

```

8.4.2025 12:39:58

merge_sort.py

Page 2/2

```

58
59 if __name__ == '__main__':
60
61     my_list = [7, 2, 9, 4, 3, 8, 6, 1]
62     original_list = my_list.copy()
63     print(my_list)
64
65     my_list = merge_sort(my_list)
66
67     print(my_list)
68     verify(original_list, my_list)
69
70     # Makeing some test to measure the time needed of merge_sort().
71     # Creating int-lists, beginning with length of 2^min_exponent
72     # until the last array with length of 2^max_exponent.
73     min_exponent = 13
74     max_exponent = 18
75     last_time = sys.float_info.max
76     for exp in range(min_exponent, max_exponent + 1):
77         length = pow(2, exp)
78         MEASUREMENTS = 10
79         min_time = sys.float_info.max
80         for i in range(MEASUREMENTS):
81             data = list(range(length))
82             random.shuffle(data)
83             #list.reverse(data)
84             start = time()
85             merge_sort(data)
86             end = time()
87             time_spent = end - start
88             if(time_spent < min_time):
89                 min_time = time_spent
90             print(f"List-Size: {length:d}          Time: {min_time*1e3:7.1f} ms          Rati
o to last: {min_time / last_time:.1f}")
91             last_time = min_time
92
93
94     """ Session-Log:
95
96     [7, 2, 9, 4, 3, 8, 6, 1]
97     [1, 2, 3, 4, 6, 7, 8, 9]
98     List-Size: 8,192          Time: ????.? ms          Ratio to last: 0.0
99     List-Size: 16,384        Time: ????.? ms          Ratio to last: ??
100    List-Size: 32,768         Time: ????.? ms          Ratio to last: ??
101    List-Size: 65,536         Time: ????.? ms          Ratio to last: ??
102    List-Size: 131,072        Time: ????.? ms          Ratio to last: ??
103    List-Size: 262,144        Time: ????.? ms          Ratio to last: ??
104
105     """

```