MergeSort

Anzahl und Reihenfolge der Aufrufe der Funktionen **`sort`** und **`merge`**

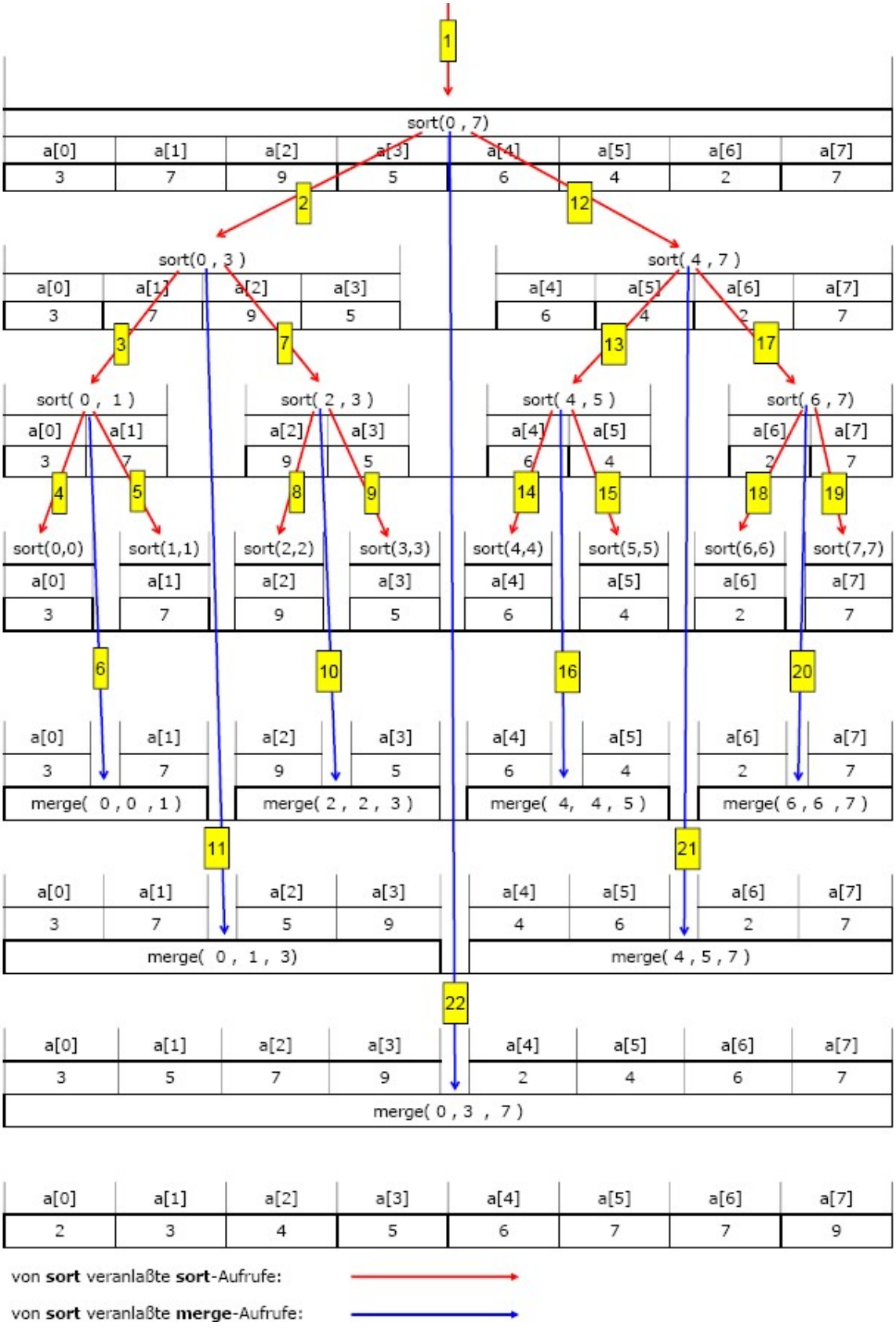
Quelltext

```
# MergeSort
# Ausgabe der Reihenfolge der Funktionsaufrufe
from random import randint
z = 0
y = 0
n = int(input('Laenge des arrays: '))
print()

# Erzeugen des arrays mit dem Namen a
# und den n Komponenten a[0], . . . , a[n-1]
a = list(range(0,n))

# Zuweisung von Zufallszahlen an die Komponenten des arrays a
for i in range(0,n): a[i] = randint(0,99)

# Ausgabe der Quelliste
anzahl = int(input('Wieviele Elemente sollen angezeigt werden? '))
print()
for i in range(0,anzahl): print(a[i])
print()

def merge(array, left, middle, right):
    global y
    y += 1
    left_sublist = array[left:middle + 1]
    right_sublist = array[middle+1:right+1]
    left_sublist_index = 0
    right_sublist_index = 0
    sorted_index = left
    while left_sublist_index < len(left_sublist) and right_sublist_index < len(right_sublist):
        if left_sublist[left_sublist_index] <= right_sublist[right_sublist_index]:
            array[sorted_index] = left_sublist[left_sublist_index]
            left_sublist_index = left_sublist_index + 1
        else:
            array[sorted_index] = right_sublist[right_sublist_index]
            right_sublist_index = right_sublist_index + 1
        sorted_index = sorted_index + 1
    while left_sublist_index < len(left_sublist):
        array[sorted_index] = left_sublist[left_sublist_index]
        left_sublist_index = left_sublist_index + 1
        sorted_index = sorted_index + 1
    while right_sublist_index < len(right_sublist):
        array[sorted_index] = right_sublist[right_sublist_index]
        right_sublist_index = right_sublist_index + 1
        sorted_index = sorted_index + 1

def sort(array, left, right):
    global z
    z +=1
    if left == right: return
    middle = (left + right)//2
    print('sort(',left,',',middle,')')
    sort(array, left, middle)
    print('sort(',middle + 1,',',right,')')
    sort(array, middle + 1, right)
    print('merge(',left,',',middle,',',right,')')
    merge(array, left, middle, right)

l = 0
r = len(a)-1
print('sort(',l,',',r,')')
sort(a, l, r)

print()
print('Sortierte Liste:')
print()
for i in range(0,anzahl): print(a[i])

print()
print('# Aufrufe sort:  ',z)
print('# Aufrufe merge: ',y)
```

Durchführung von MergeSort und Auflistung der Aufrufe der Funktionen **`sort`** und **`merge`** für eine aus den 8 Komponenten **`a[0], . . . , a[7]`** bestehende Liste **`a`**:

```
Wieviele Elemente sollen angezeigt werden? 8

89
37
31
0
19
86
33
1

sort( 0 , 7 )
sort( 0 , 3 )
sort( 0 , 1 )
sort( 0 , 0 )
sort( 1 , 1 )
merge( 0 , 0 , 1 )
sort( 2 , 3 )
sort( 2 , 2 )
sort( 3 , 3 )
merge( 2 , 2 , 3 )
merge( 0 , 1 , 3 )
sort( 4 , 7 )
sort( 4 , 5 )
sort( 4 , 4 )
sort( 5 , 5 )
merge( 4 , 4 , 5 )
sort( 6 , 7 )
sort( 6 , 6 )
sort( 7 , 7 )
merge( 6 , 6 , 7 )
merge( 4 , 5 , 7 )
merge( 0 , 3 , 7 )

Sortierte Liste:

0
1
19
31
33
37
86
89

# Aufrufe sort:   15
# Aufrufe merge:  7
```

Baumstruktur mit Reihenfolge für die Funktionsaufrufe:



08.12.2023