

## Datenstrukturen und effiziente Algorithmen Übung 9 - Aufgabe 3c

c)

### Tabelle für HeapSort-Algorithmus:

Testinstanz	1. ZM*	2. ZM	3. ZM	4. ZM	5. ZM	6. ZM	7. ZM	8. ZM	9. ZM	10. ZM	$\overline{ZM}$
desc_500000.txt	179	187	187	184	179	185	184	187	189	180	165,7
desc_600000.txt	225	214	218	217	218	220	214	219	220	217	218,2
desc_700000.txt	249	248	246	254	248	253	251	255	245	245	249,4
desc_800000.txt	287	289	285	368	349	294	289	286	290	287	302,4
desc_900000.txt	319	314	316	316	315	314	318	319	318	318	316,7
desc_1000000.txt	378	382	395	365	352	380	359	350	374	360	369,5

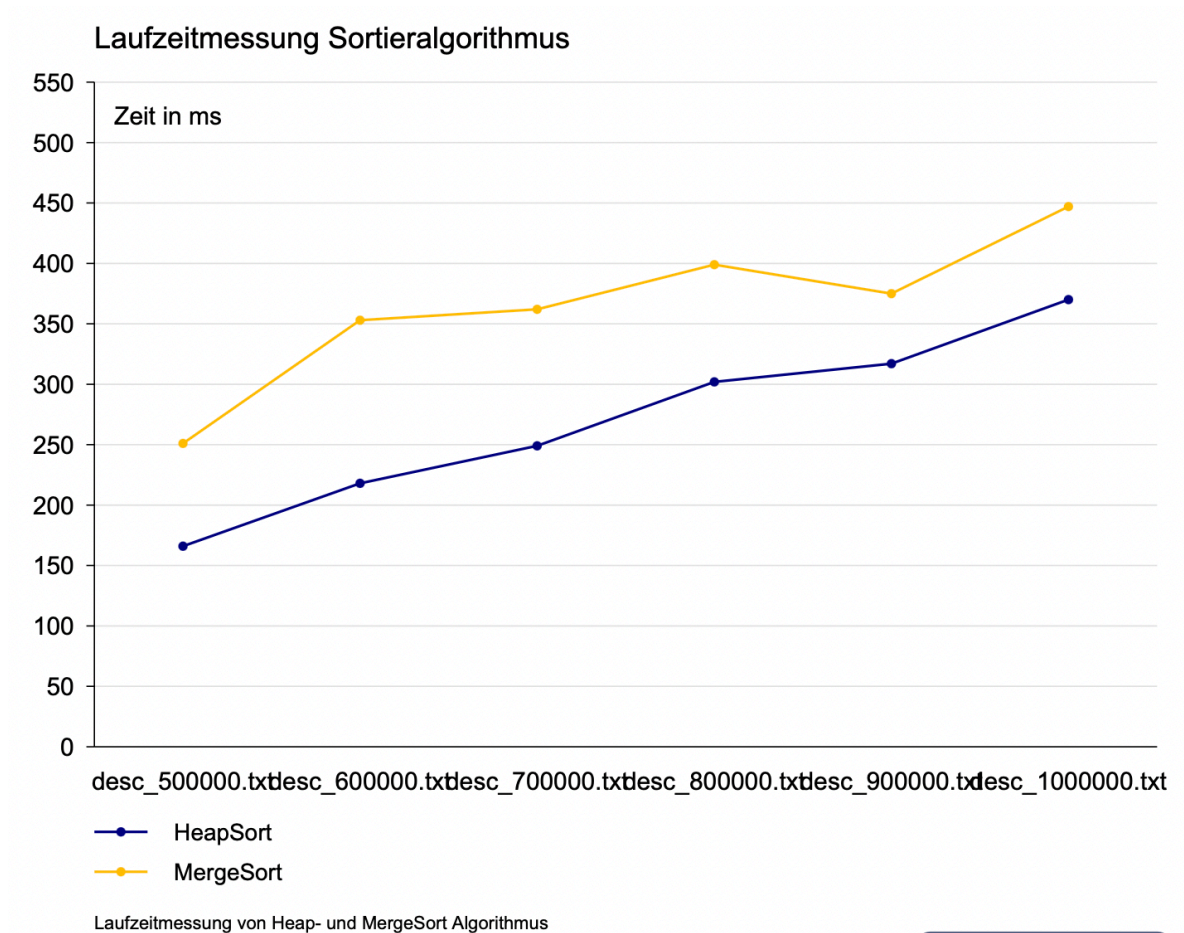
ZM\* steht für Zeitmessung in ms

### Tabelle für MergeSort-Algorithmus:

Testinstanz	1. ZM*	2. ZM	3. ZM	4. ZM	5. ZM	6. ZM	7. ZM	8. ZM	9. ZM	10. ZM	$\overline{ZM}^*$
desc_500000.txt	204	261	260	263	249	269	279	265	204	257	251,1
desc_600000.txt	331	355	432	357	353	299	297	351	394	360	352,9
desc_700000.txt	273	276	369	408	450	385	278	346	415	419	361,9
desc_800000.txt	417	413	394	394	326	446	380	443	384	394	399,1
desc_900000.txt	378	339	399	336	438	334	380	389	342	419	375,4
desc_1000000.txt	450	393	439	457	478	387	456	459	450	505	447,4

ZM\* steht für Zeitmessung in ms

$\overline{ZM}^*$  steht für die durchschnittliche Zeit in ms



Der HeapSort Algorithmus ist um einiges schneller als der MergeSort-Algorithmus vor allem, wenn die Größe der zu sortierenden Zahlenfelder größer werden, dann schneidet HeapSort besser ab (siehe Tabelle Seite 1).