b. sort works correctly.

[20, 98, 33, 87, 73, 7, 23, 51, 54, 11]

[7, 11, 20, 23, 33, 51, 54, 73, 87, 98]

c.

|  |  |  |
| --- | --- | --- |
| Input Size | Heap sort running time | Merge sort running time |
| 4 | 4700 | 4600 |
| 16 | 14800 | 2200 |
| 64 | 74000 | 3900 |
| 256 | 163200 | 11900 |
| 1024 | 263700 | 45200 |
| 4096 | 769900 | 138600 |
| 16384 | 2201900 | 541900 |
| 65536 | 8555300 | 1240500 |
| 262144 | 25916900 | 1750600 |
| 1048576 | 108074600 | 3190200 |
| 4194304 | 480481800 | 15251000 |
| 16777216 | 1785095700 | 43024800 |
| 67108864 | 7590090900 | 154320600 |

d.

**2.138.952 KB** memory is used (working set column) for heap sort.

e.

**2.140.812 KB** memory is used (working set column) for merge sort.

It shows us that merge sort uses more ram. The reason for this is that merge sort is not in-place algorithm which means that it requires extra memory space. Since while implementing merge sort, we need to create 2 sub-arrays to able to sort the original array. To conclude that, merge sort consumes more memory compared to heap sort.