Merge Sort Report

CS6301.g42

David Tan, Khaled Al-naami, Peter Farago, Yu Lin

The running times for different data sizes and algorithms are shown below. These running times represent averages taken over several runs of each algorithm for each data size.

Alg. / ***Size***	**MergeSort** V1 (milliseconds)	**MergeSort** V2 (milliseconds)	**MergeSort** V3 (milliseconds)	**MergeSort** V4 (milliseconds)
0.1 M	3099	20	24	20
0.5 M	n/a	88	73	66
1 M	n/a	177	135	131
5 M	n/a	922	735	727
10 M	n/a	1801	1456	1420

Table 1 Average Running Time of different version of MergSort

Basically, the first version cannot work when the data size is larger than 0.5M. Version 2, Version 3 and Version 4 works relatively well. Version 2 and Version 3 has little different while the data size is small. However, when the data size achieves 10M, Version 2 need quite more time than Version 3.

There is another more interesting point. From a quick glance we can see the version 3 and version 4 finish their task almost at the same time. However, according to the pseudo code, version 4 should be much faster than version 3 due to no array copy operation. One possible reason is maybe the arraycopy() methods is an optimized copy methods that cost less time than naïve copy method.