Implementation Analysis

Efficiency – I ran the program with 2 and all the way up to 10 threads to compare with the single-thread serial mergesort (input size = 100,000,000 elements), and here is a chart that shows the speed efficiency:

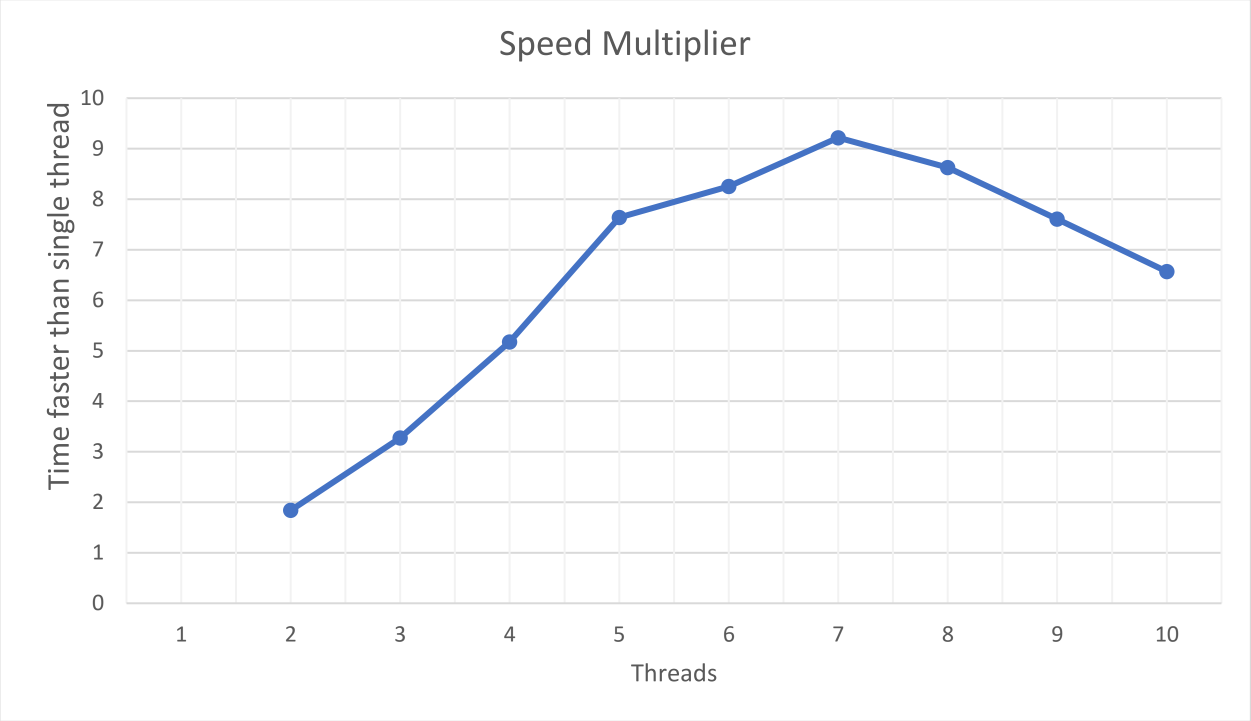

Figure 1: Efficiency chart

The speedup seems to increase linearly as the number of threads are increased up until 7 threads, then with the higher number of threads, it starts to slow down probably because it uses more threads than number of CPU cores available which cause threads to fight over the CPU resource.

Effectiveness – here is a table of data that shows the speed of serial mergesort and threaded mergesort with elements in the range from 1 million to 100 million:

# of elements	1M	2M	3M	4M	5M	6M	7M	8M	9M	10M	50M	100M
Serial	0.26	0.54	0.89	1.16	1.52	1.89	2.03	2.41	2.7	3.1	17.11	35.43
Threaded	0.05	0.11	0.16	0.21	0.27	0.33	0.36	0.42	0.43	0.54	2.73	5.31

Figure 2: Effectiveness table

The threaded mergesort always outperforms the serial version as we can see.