Assignment 3

Garvit Shah [U21CS089]

February 2023

Hardware Details

- Memory: 8 GB 1600 MHz DDR3
- Processor: 1.8 GHz Dual-Core Intel Core i5

Software Details

- Apple clang version 14.0.0 clang-1400.0.29.202
- xcode-select version 2395.

1 2-Way Merge Sort

1.1 Algorithm

1. Create a file pointer and Open the File
2. Read the number from the file and put it in an array.
3. Divide the array into two equal parts recursively till you are left with one element.
4. After reaching the base case, call merge function.
5. Merge Funciton merges two divided arrays.

1.2 Observations

Time Complexity for 2-way Merge Sort		
File Name	No. of Entries	Average Case
File-1	1024	0.00379398
File-2	4096	0.0158169
File-3	16384	0.0653478
File-4	65536	0.271358
File-5	262144	1.13362
File-6	1048576	5.25466
File-7	2097152	10.6776
File-8	4194304	21.6901
File-9	8388608	44.4687
File-10	16777216	72.4756

Table 1: Time taken for 2-way Merge Sort

2 3-Way Merge Sort

2.1 Algorithm

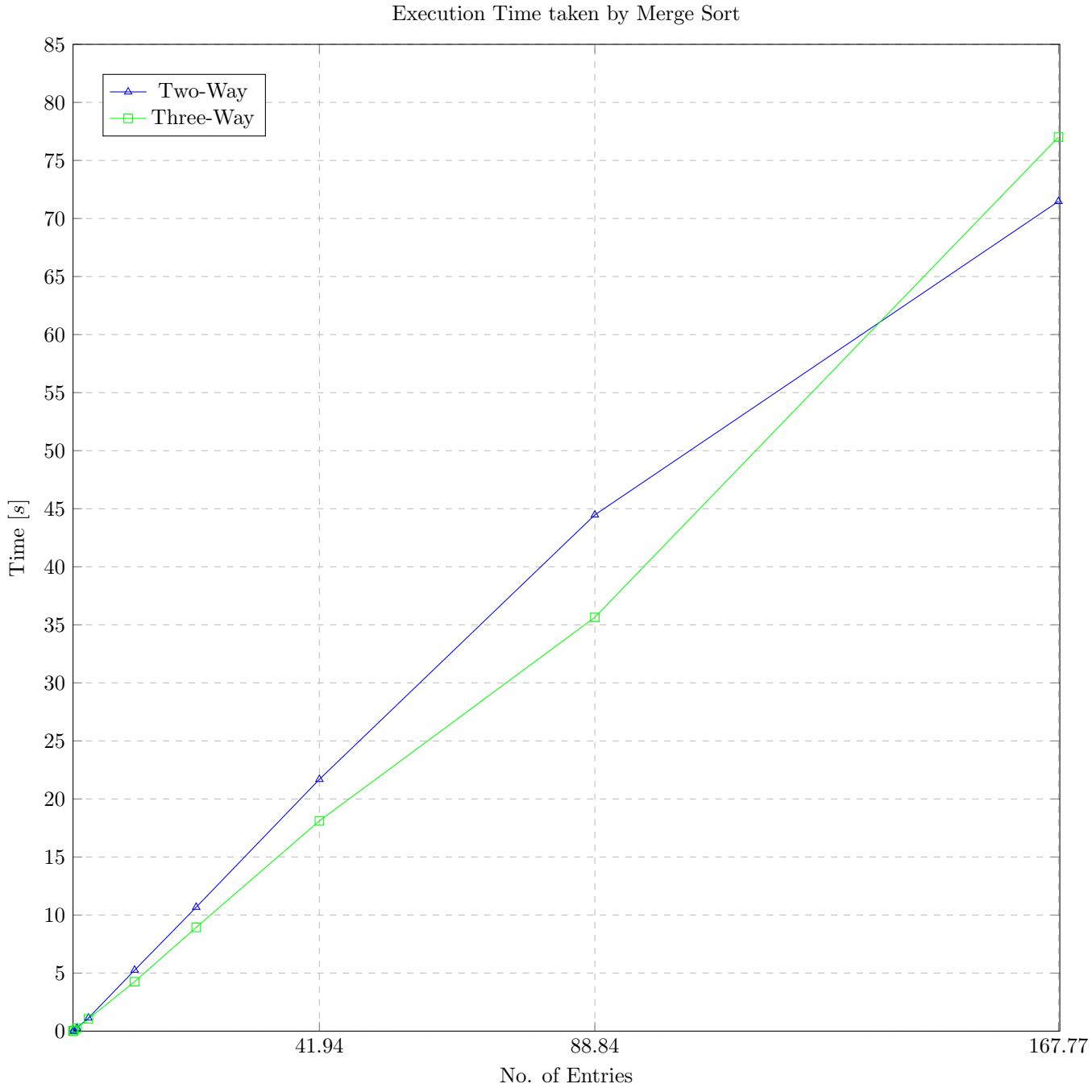
1. Create a file pointer and Open the File
2. Read the number from the file and put it in an array.
3. Divide the array into three equal parts recursively till you are left with one element in each divided array.
4. After reaching the base case, call merge function.
5. Merge Funciton merges three divided arrays.

2.2 Observations

Time Complexity for 3-way Merge Sort		
File Name	No. of Entries	Average Case
File-1	1024	0.00379398
File-2	4096	0.0158169
File-3	16384	0.0653478
File-4	65536	0.271358
File-5	262144	1.13362
File-6	1048576	5.25466
File-7	2097152	10.6776
File-8	4194304	21.6901
File-9	8388608	44.4687
File-10	16777216	72.4756

Table 2: Time taken for 3-way Merge Sort

2.3 Graph



2.4 Conclusion

The graph for the worst case is of type *nlogn*. This type of curve implies that the change in time taken is linearly and logarithimically dependent on the change in the no. of elements in the file, as the line is given by $y = xlog(x)$. Therefore, it can be concluded that the time complexity for Merge Sort is $O(nlog(n))$. Theoretically also time complexity comes out to be $O(nlog(n))$. Thus the conclusion matches to the theoretical value of the time complexity.

Time Complexity = $\theta(nlog(n))$