Name: Mani Charan Reddy Loka
NUID: 002727403

Task:
To determine--for sorting algorithms--what is the best predictor of total execution time: comparisons, swaps/copies, hits (array accesses), or something else.

Relationship Conclusion:

By running the experiments for different combinations of cutoff values, threads, and array sizes, sufficient data is generated to conclude the optimal number of threads required for the algorithm. Upon analyzing the same, it can be seen that using 4 threads gives the best performance while increasing it any further does not have a significant improvement in the performance as the cost of thread creation and synchronization might offset the work done by parallelism.

On average, the lowest runtime is achieved when the cutoff value is 30% of the array size.

The relationship between thread count(t) and recursion depth d is $t=2^d$

Evidence to Conclusion:

Below are the runtimes in ms for different combinations of array size, threads, and cutoffs

ArraySize: 100000

Cutoff\No of Threads	2	4	8	16	32	64
10000	293ms	123ms	85ms	66ms	61ms	67ms
20000	163ms	73ms	62ms	60ms	62ms	59ms
30000	141ms	104ms	66ms	60ms	63ms	63ms
40000	112ms	65ms	65ms	65ms	62ms	64ms
50000	97ms	65ms	63ms	61ms	62ms	62ms
60000	86ms	84ms	79ms	84ms	80ms	80ms
70000	87ms	84ms	83ms	83ms	84ms	79ms
80000	123ms	84ms	81ms	80ms	83ms	82ms
90000	88ms	79ms	80ms	79ms	81ms	82ms
100000	88ms	89ms	82ms	78ms	84ms	76ms

Time in ms vs cut-off graph for array size of 100000.
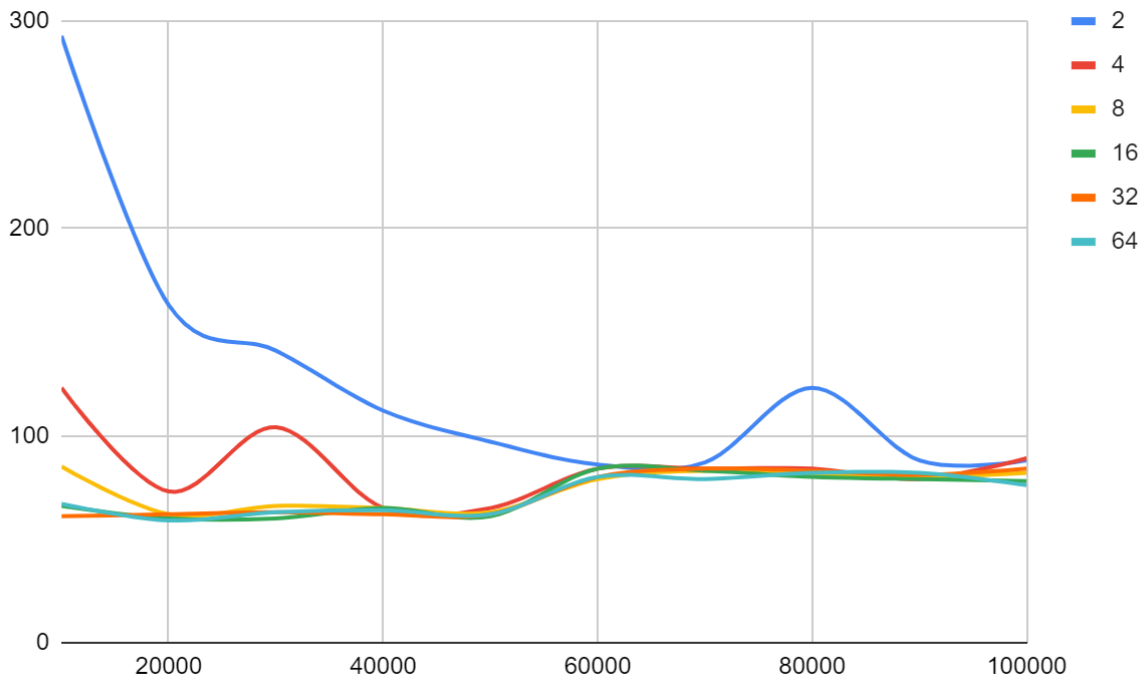
Program Structures and Algorithms
Spring 2023(Section - 1)

Below are the values for array size 250000

Cutoff\No of Threads	2	4	8	16	32	64
10000	489.00	263.00	178.00	181.00	131.00	171.00
20000	389.00	158.00	137.00	135.00	126.00	144.00
30000	241.00	158.00	134.00	129.00	151.00	129.00
40000	199.00	159.00	143.00	131.00	127.00	140.00
50000	231.00	151.00	139.00	147.00	130.00	147.00
60000	198.00	154.00	134.00	131.00	123.00	135.00
70000	232.00	153.00	143.00	145.00	165.00	147.00
80000	332.00	158.00	145.00	148.00	168.00	144.00
90000	272.00	147.00	150.00	140.00	157.00	140.00
100000	243.00	147.00	149.00	138.00	190.00	143.00

Program Structures and Algorithms
Spring 2023(Section - 1)

Time in ms vs cutoff graph for array size = 250000
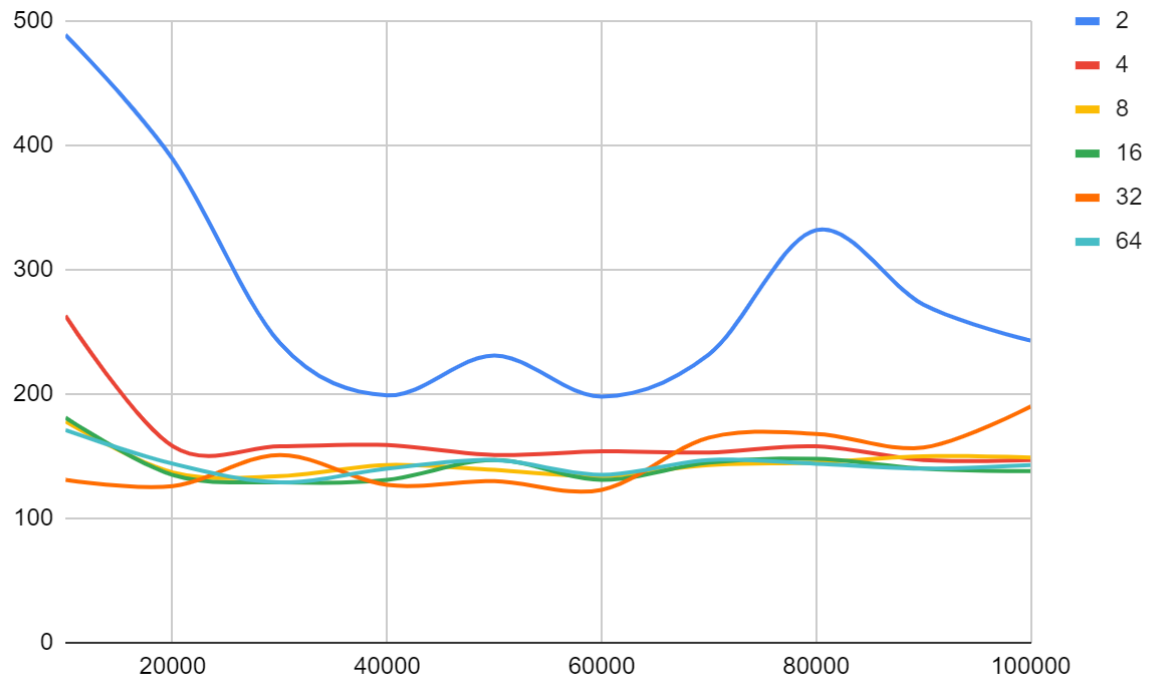
Below are the values for array size = 500000

Cutoff\No of Threads	2	4	8	16	32	64
10000	961.00	396.00	306.00	315.00	361.00	295.00
20000	346.00	322.00	247.00	289.00	281.00	286.00
30000	368.00	280.00	253.00	289.00	285.00	277.00
40000	394.00	290.00	260.00	293.00	286.00	290.00
50000	353.00	291.00	282.00	304.00	290.00	298.00
60000	360.00	299.00	281.00	316.00	311.00	315.00
70000	417.00	305.00	275.00	311.00	314.00	311.00
80000	411.00	297.00	257.00	309.00	298.00	291.00
90000	410.00	296.00	259.00	301.00	299.00	289.00
100000	406.00	299.00	255.00	301.00	293.00	280.00

Time in ms vs cutoff graph for array size = 500000
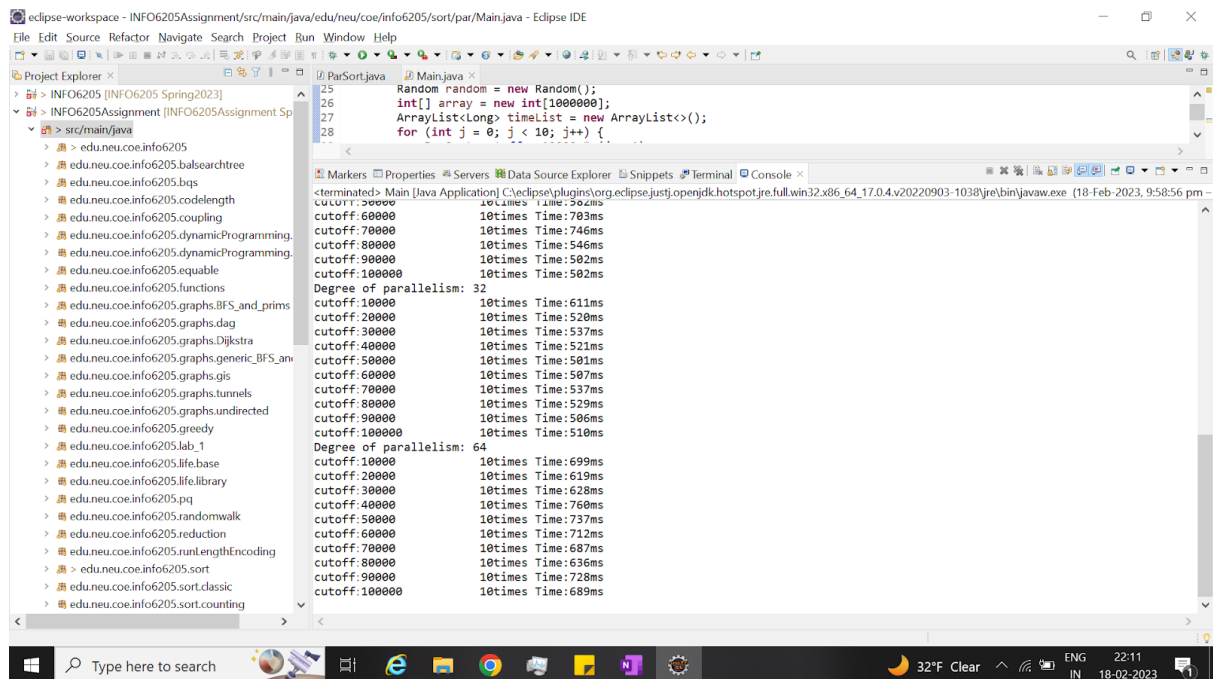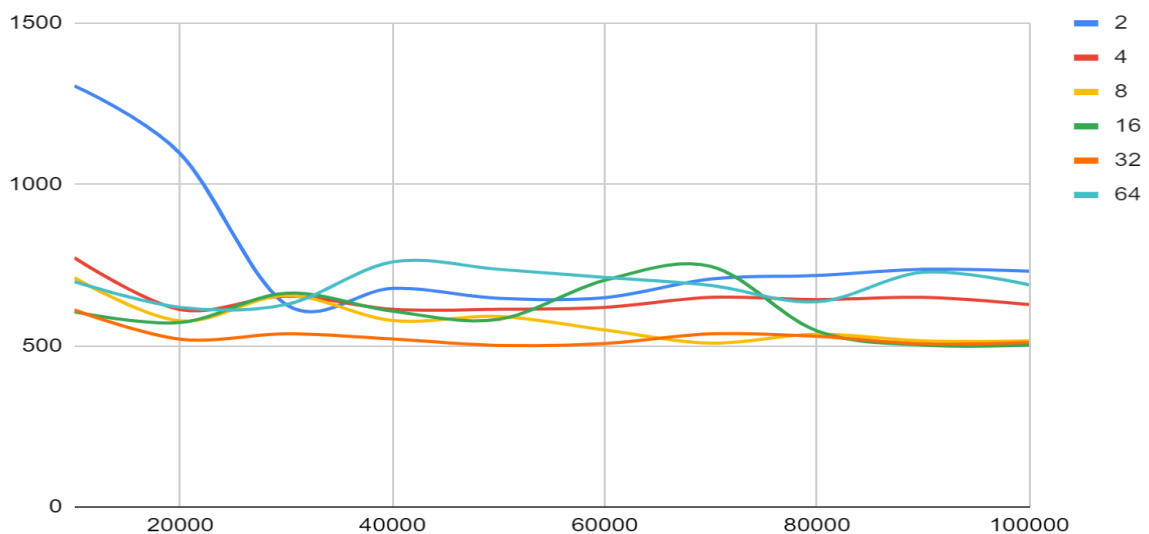
Program Structures and Algorithms
Spring 2023(Section - 1)

Below are the values for array size = 1000000

Cutoff\No of Threads	2	4	8	16	32	64
10000	1,306.00	773.00	711.00	605.00	611.00	699.00
20000	1,095.00	612.00	577.00	572.00	520.00	619.00
30000	627.00	654.00	656.00	663.00	537.00	628.00
40000	678.00	613.00	578.00	607.00	521.00	760.00
50000	647.00	613.00	591.00	582.00	501.00	737.00
60000	649.00	619.00	549.00	703.00	507.00	712.00
70000	707.00	650.00	508.00	746.00	537.00	687.00
80000	718.00	643.00	536.00	546.00	529.00	636.00
90000	737.00	650.00	515.00	502.00	506.00	728.00
100000	731.00	628.00	515.00	502.00	510.00	689.00

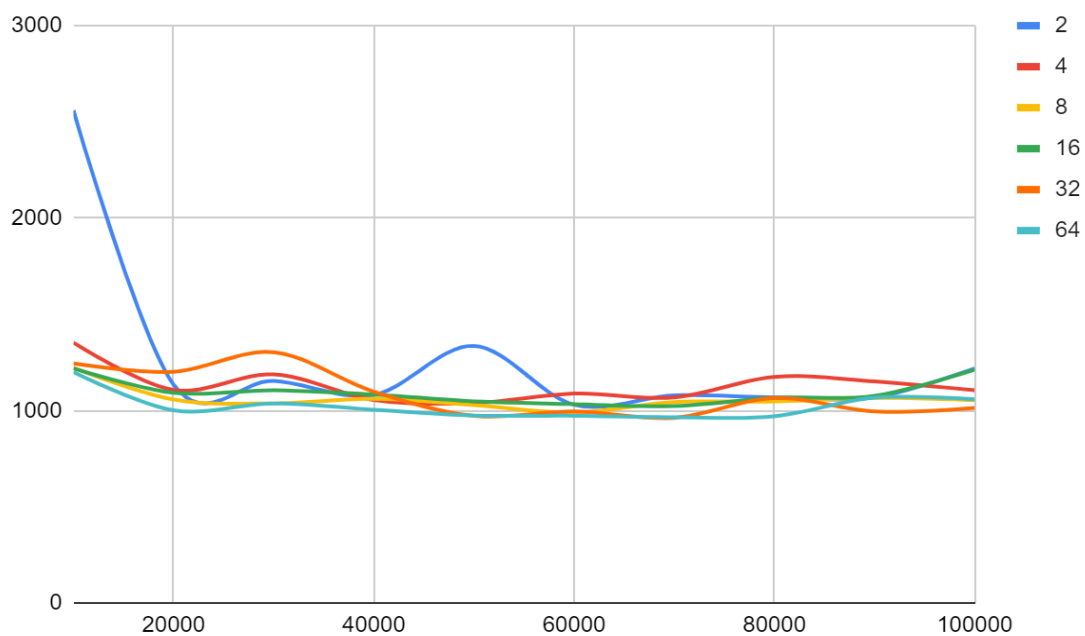Time in ms vs cutoff graph for array size = 1000000

Program Structures and Algorithms
Spring 2023(Section - 1)

Below are the values for array size = 2000000

Cutoff\No of Threads	2	4	8	16	32	64
10000	2,560.00	1,353.00	1,222.00	1,219.00	1,245.00	1,200.00
20000	1,134.00	1,107.00	1,057.00	1,092.00	1,201.00	1,001.00
30000	1,154.00	1,188.00	1,037.00	1,105.00	1,303.00	1,037.00
40000	1,081.00	1,055.00	1,061.00	1,083.00	1,098.00	1,004.00
50000	1,336.00	1,039.00	1,028.00	1,048.00	972.00	974.00
60000	1,030.00	1,088.00	990.00	1,033.00	995.00	973.00
70000	1,079.00	1,068.00	1,044.00	1,023.00	962.00	965.00
80000	1,068.00	1,175.00	1,047.00	1,066.00	1,065.00	970.00
90000	1,069.00	1,151.00	1,066.00	1,076.00	995.00	1,067.00
100000	1,219.00	1,105.00	1,054.00	1,213.00	1,013.00	1,058.00

The graphs show that increasing the thread count doesn't significantly improve the sort time of the parallel sort. The sort time is optimal when the value of thread is 4. Hence, using 4 threads is the optimal way of using parallel sorting.

Also, upon comparing the cutoff at which the which the 4 thread time is optimal we can find out that it is always optimal when the cutoff length is about 20~40% of the input length. Hence, we can average the value to around 30%.

Below is the screenshot of csv files generated for every different combination of array size and thread count

ArraySize100000Threads32result.csv	18-02-2023 21:40	XLS Worksheet
ArraySize100000Threads64result.csv	18-02-2023 21:40	XLS Worksheet
ArraySize250000Threads2result.csv	18-02-2023 21:40	XLS Worksheet
ArraySize250000Threads4result.csv	18-02-2023 21:40	XLS Worksheet
ArraySize250000Threads8result.csv	18-02-2023 21:40	XLS Worksheet
ArraySize250000Threads16result.csv	18-02-2023 21:40	XLS Worksheet
ArraySize250000Threads32result.csv	18-02-2023 21:40	XLS Worksheet
ArraySize250000Threads64result.csv	18-02-2023 21:40	XLS Worksheet
ArraySize500000Threads2result.csv	18-02-2023 21:50	XLS Worksheet
ArraySize500000Threads4result.csv	18-02-2023 21:50	XLS Worksheet
ArraySize500000Threads8result.csv	18-02-2023 21:50	XLS Worksheet
ArraySize500000Threads16result.csv	18-02-2023 21:50	XLS Worksheet
ArraySize500000Threads32result.csv	18-02-2023 21:50	XLS Worksheet
ArraySize500000Threads64result.csv	18-02-2023 21:50	XLS Worksheet
ArraySize1000000Threads2result.csv	18-02-2023 21:59	XLS Worksheet
ArraySize1000000Threads4result.csv	18-02-2023 21:59	XLS Worksheet
ArraySize1000000Threads8result.csv	18-02-2023 21:59	XLS Worksheet
ArraySize1000000Threads16result.csv	18-02-2023 21:59	XLS Worksheet
ArraySize1000000Threads32result.csv	18-02-2023 21:59	XLS Worksheet
ArraySize1000000Threads64result.csv	18-02-2023 21:59	XLS Worksheet