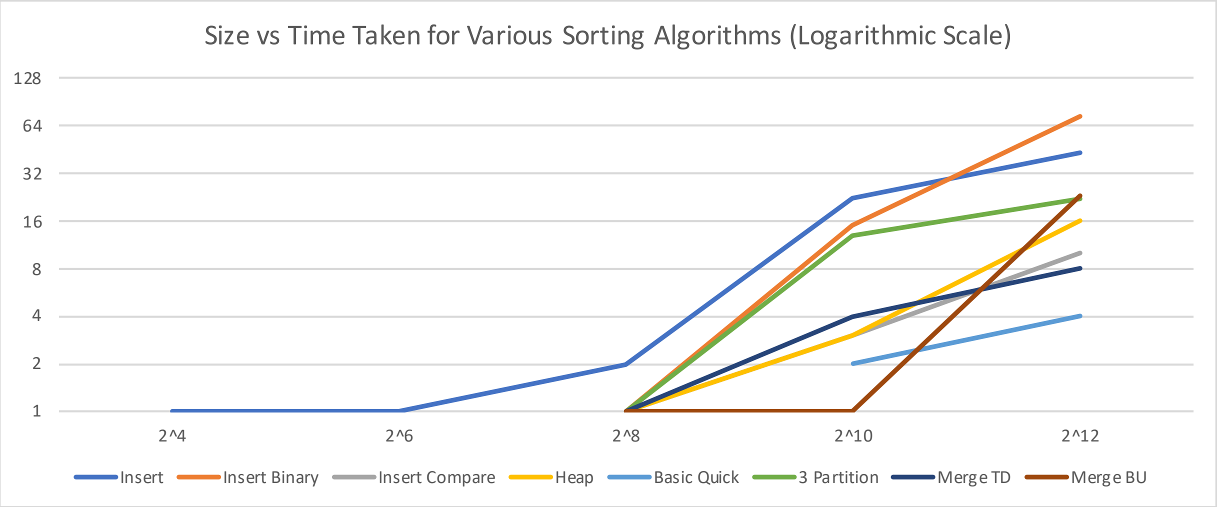
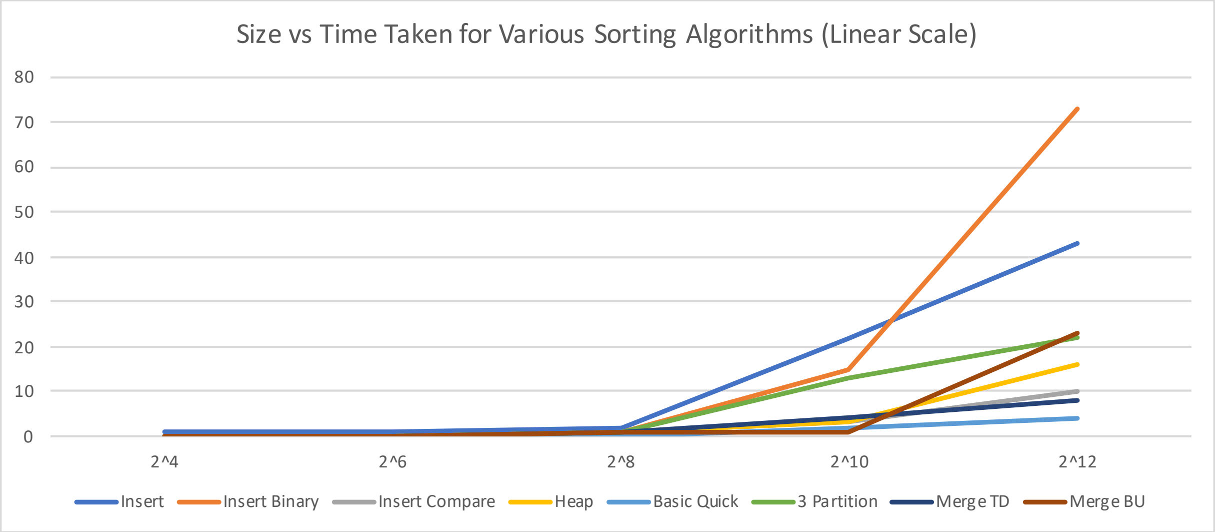
	time (ms)	time (ms)	time (ms)	time (ms)	time (ms)	time (ms)	time (ms)	time (ms)
Size	**Insert**	**Insert Binary**	**nsert Compar**	**Heap**	**Basic Quick**	**3 Partition**	**Merge TD**	**Merge BU**
2^4	1	0	0	0	0	0	0	0
2^6	1	0	0	0	0	0	0	0
2^8	2	1	1	1	0	1	1	1
2^10	22	15	3	3	2	13	4	1
2^12	43	73	10	16	4	22	8	23





Since some of the algorithms sometimes produce drastically different times, multiple different timings were considered when determining the time complexities for each algorithms. Also note that since some values were too small to be counted by the stopwatch, they apear as zero, which does **not** apear on the log scale graph as log(0) is undefined. Milliseconds were used instead of seconds to avoid negative numbers (log of numbers below 1).

Hypothesis

Algorithm:	Basic Quick	Merge TD	Merge BU	Insertion	Insert-Binary	Insert Compare	3P Quick	Heap
Time Complexity:	Nlog(N)	Nlog(N)	Nlog(N)	N^2	N^2	N^2	Nlog(N)	Nlog(N)

Most of the determined time complexities are expected, except for binary insertion. It was originally expected to take Nlog(N). It's N^2 complexity could be due to the number of swaps still required for each single insertion. Furthermore, insert Comparable is the same as insertion, except without using the comparable interface. This difference should only effect the coefficient which is negligible in big O notation, which is why the two algorithms have the same time complexity.

Predictions (time in ms):

Algorithm:	Basic Quick	Merge TD	Merge BU	Insertion	Insert Binary	Insert Compare	3P Quick	Heap
2^14:	9	14	33	100	100	100	40	30
2^16:	20	25	45	1000	1000	1000	80	60

Results (time in ms):

Algorithm:	Basic Quick	Merge TD	Merge BU	Insertion	Insert Binary	Insert Compare	3P Quick	Heap
2^14:	11	13	24	144	119	242	44	22
2^16:	22	40	26	2884	2020	4614	45	30

According to results, predictions were somewhat accurate, meaning original hypothesis (based on first set of results) is also somewhat accurate in describing the time complexities of the algorithms.