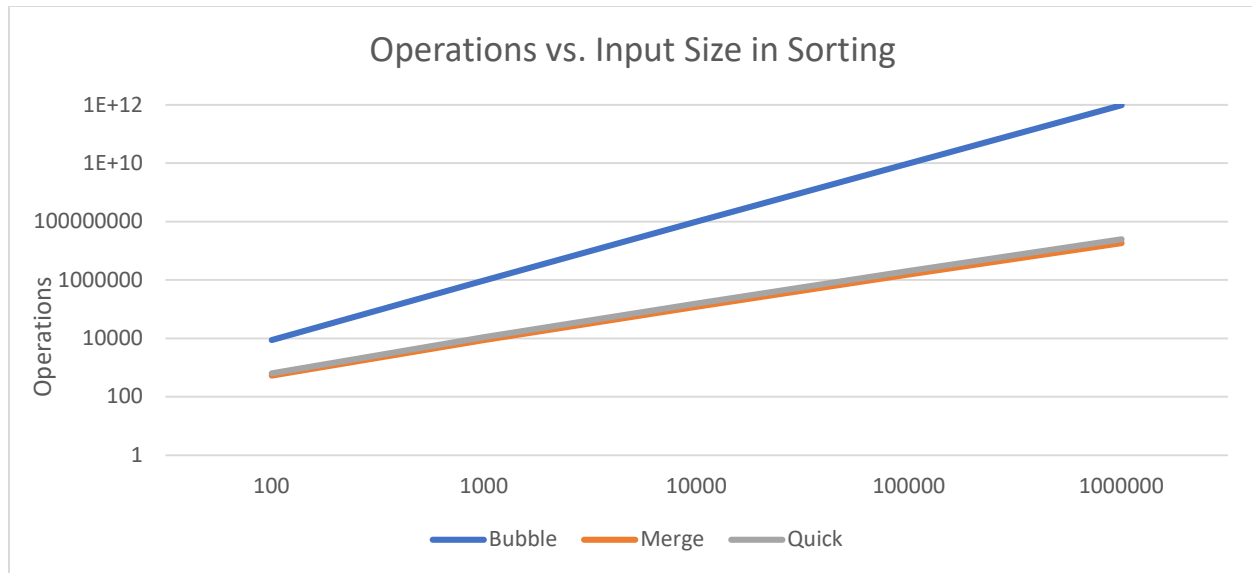


Question 5 – Analyzation of Efficiency of Sorting Algorithms – Alexander Sun

Introduction of Algorithms:

Comparing Bubble sort ($O(n^2)$), merge sort ($O(n \log n)$), and quick sort ($O(n \log n)$).

Graphs & Data:



Raw Data:

Operations per Input Size of Sorting Algorithms				
Input Size	Bubble	Merge	Quick	
100	8845	541	645	
1000	961195	8707	10988	
10000	98805918	120451	155605	
100000	9761525892	1536365	2019394	
1000000	9.75475E+11	18674241	24786385	

Observations/Reflection:

My initial observation is that Merge sort & Quick sort operations come very close to $(n \log n)$. Merge sort based on operations does less than quick sort (though not faster b/c memory & other constraints). My theory as to why this occurs is because when one half has completed merging and the other hasn't, no more comparisons need to occur, so it skips out on some of the operations that are counted. Furthermore, the partition that I implemented quick sort compares every value from high to low so all the operations are counted. Merge sort also does a lot of copying that isn't counted as it isn't a comparison. Clearly Bubble sort is much much slower as it's an $O(n^2)$ algorithm.