| Objects Generated | BubbleSort Time (NanoSeconds) | MergeSort Time (NanoSeconds) |
| --- | --- | --- |
| 5 | 3,625 | 16,747 |
| 100 | 407,381 | 422,702 |
| 500 | 786,714 | 593,875 |
| 10,000 | 197,565,790 | 5,564,860 |
| 100,000 | 40,558,964,147 | 19,077,946 |

\*Highlight = Quicker Method\*

* The algorithm analysis conclusions from class apply when we are sorting objects as it is evident in my observations:
* In class, we discussed how with larger amounts of data, an O(log2n) sort algorithm would be more efficient than an O(n2) sort algorithm, simply because in the O(log2n) algorithm, the data gets split in half multiple times, analyzed in such a way that the sorting can focus on multiple smaller sections, arrange them accordingly, and then combine them in the proper order that we ask for.
* I did notice, however, that when it comes to smaller data sets, O(n2) sort algorithm is way more efficient, due to the fact that MergeSort needs time to create the “rules” and “splits” for the data, which takes a certain amount of time. This time is tiny with bigger data sets, but with smaller data sets, bubble sort can go straight to work and therefore it is the quicker method.