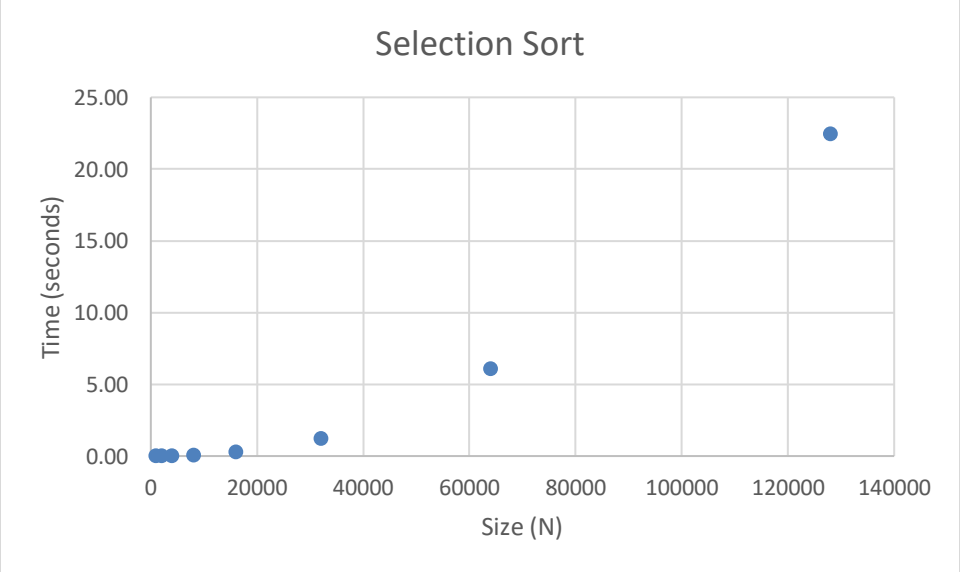
Li Xi

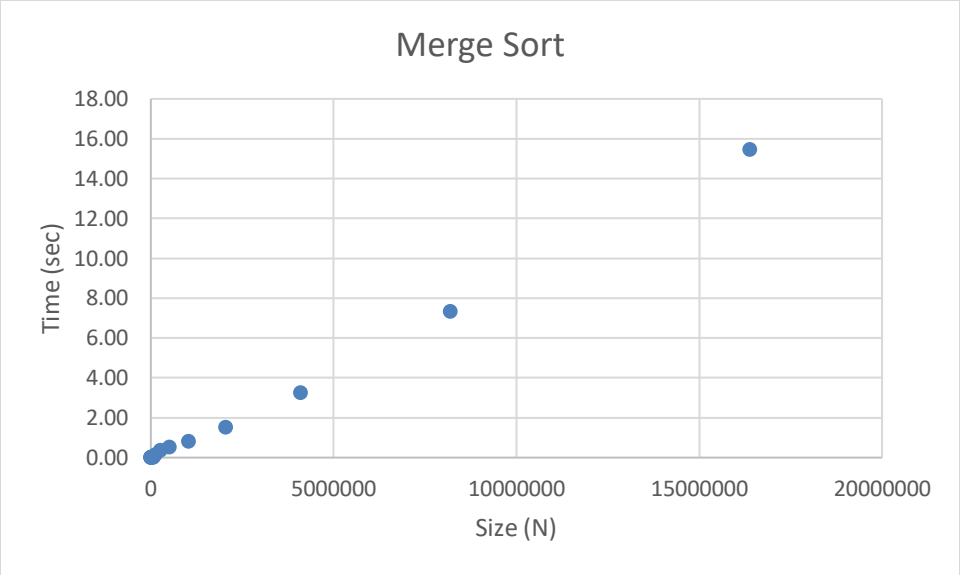
Selection sort: b = 2, therefore the growth rate of selection sort is quadratic.

| Selection sort | | |
|---|---|---|
| **Size (N)** | **Time (seconds)** | **Log Ratio (b)** |
| 1000 | 0.00 | |
| 2000 | 0.02 | #DIV/0! |
| 4000 | 0.02 | 0.00 |
| 8000 | 0.08 | 2.00 |
| 16000 | 0.30 | 1.91 |
| 32000 | 1.20 | 2.00 |
| 64000 | 6.06 | 2.34 |
| 128000 | 22.47 | 1.89 |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| b = 2 | | |



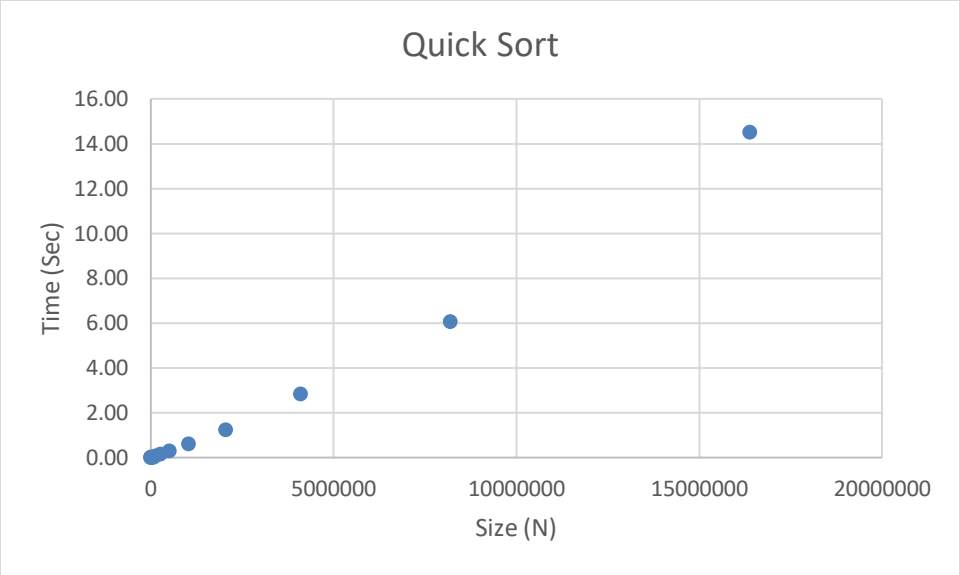
Merge Sort: this equation (Running Time = $aN^b$) is a simple one which does not explain Merge Sort. The growth rate of Merge Sort is linearithmic (NlogN), which is more than linear but less than quadratic.

| Merge sort | | |
|---|---|---|
| **Size (N)** | **Time (seconds)** | **Log Ratio (b)** |
| 1000 | 0.00 | |
| 2000 | 0.00 | #DIV/0! |
| 4000 | 0.00 | #DIV/0! |
| 8000 | 0.00 | #DIV/0! |
| 16000 | 0.00 | #DIV/0! |
| 32000 | 0.02 | #DIV/0! |
| 64000 | 0.03 | 0.58 |
| 128000 | 0.13 | 2.12 |
| 256000 | 0.38 | 1.55 |
| 512000 | 0.52 | 0.45 |
| 1024000 | 0.83 | 0.67 |
| 2048000 | 1.52 | 0.87 |
| 4096000 | 3.27 | 1.11 |
| 8192000 | 7.33 | 1.16 |
| 16384000 | 15.47 | 1.08 |
| | | |
| b = 1 | | |



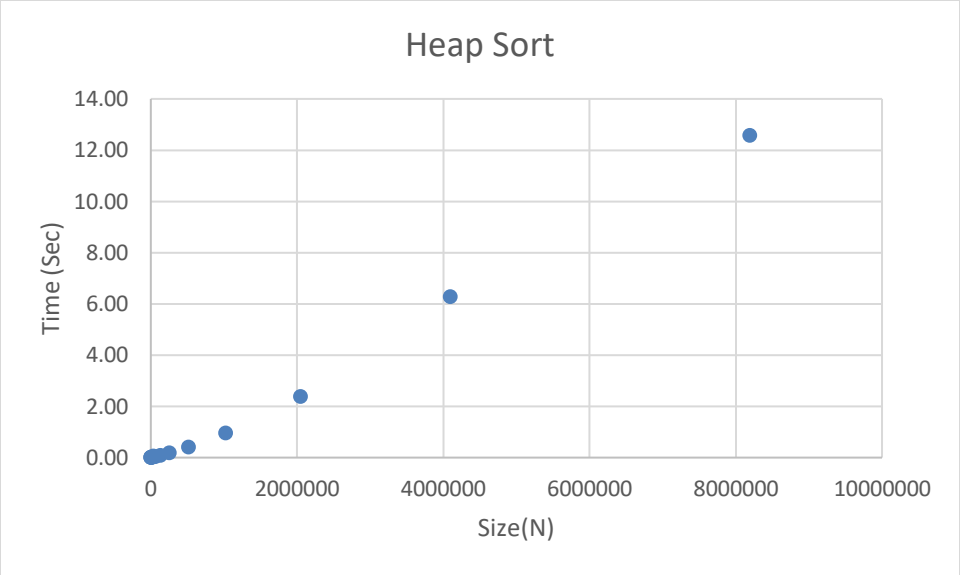

Quick Sort: Similar to Merge Sort, Quick Sort's growth rate is also linearithmic (NlogN).

| Quick sort | | |
|---|---|---|
| **Size (N)** | **Time (seconds)** | **Log Ratio (b)** |
| 1000 | 0.00 | |
| 2000 | 0.00 | #DIV/0! |
| 4000 | 0.02 | #DIV/0! |
| 8000 | 0.02 | 0.00 |
| 16000 | 0.02 | 0.00 |
| 32000 | 0.02 | 0.00 |
| 64000 | 0.02 | 0.00 |
| 128000 | 0.08 | 2.00 |
| 256000 | 0.16 | 1.00 |
| 512000 | 0.30 | 0.91 |
| 1024000 | 0.61 | 1.02 |
| 2048000 | 1.25 | 1.04 |
| 4096000 | 2.83 | 1.18 |
| 8192000 | 6.06 | 1.10 |
| 16384000 | 14.52 | 1.26 |
| | | |
| b = 1 | | |




Heap Sort: Similar to Merge Sort and Quick Sort, Heap Sort is also NlogN.

| Heap sort | | |
|---|---|---|
| **Size (N)** | **Time (seconds** | **Log Ratio (b)** |
| 1000 | 0.00 | |
| 2000 | 0.00 | #DIV/0! |
| 4000 | 0.00 | #DIV/0! |
| 8000 | 0.02 | #DIV/0! |
| 16000 | 0.00 | #NUM! |
| 32000 | 0.05 | #DIV/0! |
| 64000 | 0.03 | -0.74 |
| 128000 | 0.08 | 1.42 |
| 256000 | 0.19 | 1.25 |
| 512000 | 0.41 | 1.11 |
| 1024000 | 0.95 | 1.21 |
| 2048000 | 2.39 | 1.33 |
| 4096000 | 6.28 | 1.39 |
| 8192000 | 12.58 | 1.00 |
| | | |
| | | |
| b = 1 | | |




In conclusion, quick sort, merge sort, and heap sort are faster than selection sort. In further comparison, I found that quick sort is the fastest by comparing these three sorts based on one particular sample size 16384000. Quicksort's time running this large sample is 14.52 seconds, which is faster than merge sort's running time 15.47 seconds.