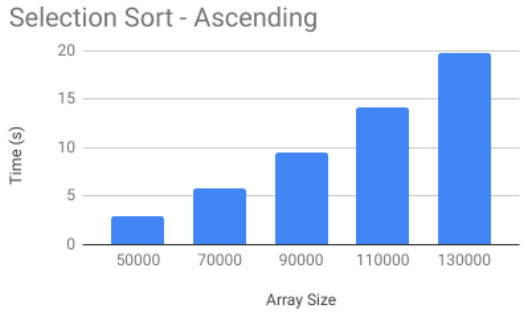
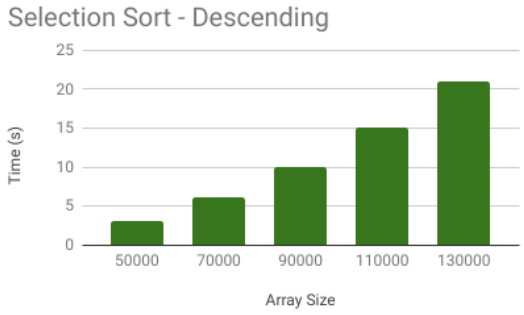
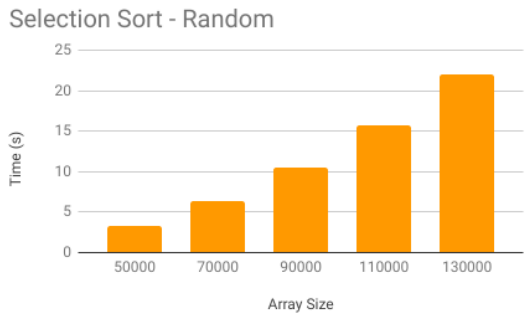
# Evan Trout
# EECS 268 Lab 7 Report

### Selection Sort

| Size | Ascending (s) | Descending (s) | Random (s) |
|---|---|---|---|
| 50000 | 2.92080 | 3.14923 | 3.26441 |
| 70000 | 5.79518 | 6.11749 | 6.41747 |
| 90000 | 9.51462 | 10.0613 | 10.5619 |
| 110000 | 14.1941 | 14.9978 | 15.7432 |
| 130000 | 19.8115 | 20.9188 | 22.0235 |




Analysis of the curves for all three cases appear to indicate that selection sort has $O(n^2)$ behavior.

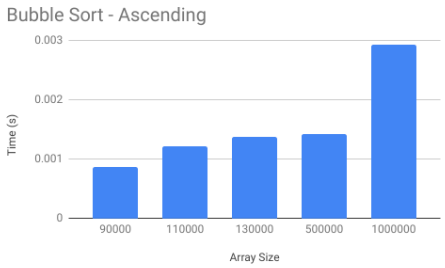
Based on this assumption, the time to sort an an array of size 10,000,000 can be estimated by solving the ratio $t_1/t_2 = O_1/O_2$:

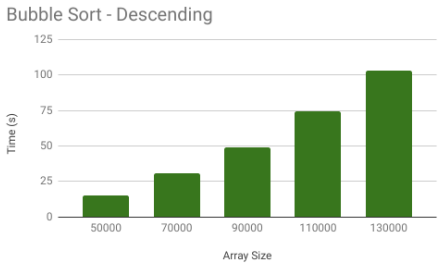
| Size | Ascending (s) | Descending (s) | Random (s) |
|---|---|---|---|
| 10000000 | 116832 | 125969 | 130576.4 |

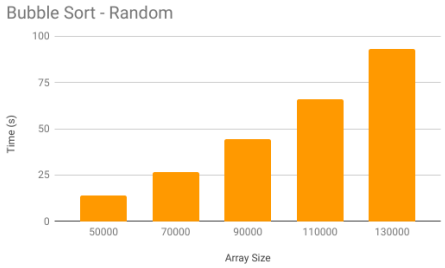
**Bubble Sort**

| Size | Ascending (s) |
|---|---|
| 90000 | 0.000874 |
| 110000 | 0.001222 |
| 130000 | 0.001373 |
| 500000 | 0.001418 |
| 1000000 | 0.002922 |

| Size | Descending (s) | Random (s) |
|---|---|---|
| 50000 | 15.269 | 14.0136 |
| 70000 | 30.4555 | 26.8356 |
| 90000 | 49.2996 | 44.3323 |
| 110000 | 74.53 | 66.1054 |
| 130000 | 102.999 | 93.0703 |




Analysis of the curves appears to indicate that the ascending case has behavior of O(n), while the descending and random cases have behavior of $O(n^2)$.

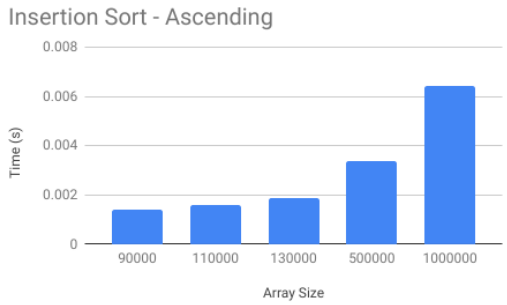
Based on this assumption, the time to sort an an array of size 10,000,000 can be estimated by solving the ratio $t_1/t_2 = O_1/O_2$:

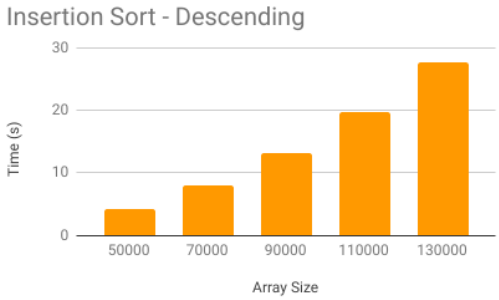
| Size | Ascending (s) | Descending (s) | Random (s) |
|---|---|---|---|
| 10000000 | 0.02836 | 610760 | 560544 |

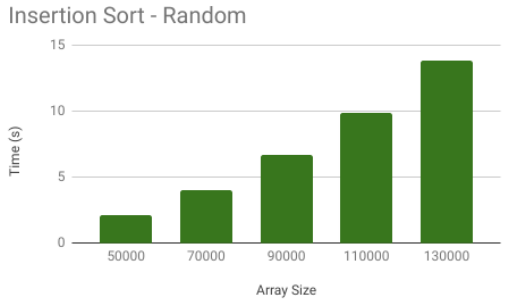
**Insertion Sort**

| Size | Ascending (s) |
|---|---|
| 90000 | 0.001425 |
| 110000 | 0.001612 |
| 130000 | 0.001866 |
| 500000 | 0.003396 |
| 1000000 | 0.006408 |

| Size | Descending (s) | Random (s) |
|---|---|---|
| 50000 | 4.14375 | 2.10959 |
| 70000 | 8.00778 | 4.0564 |
| 90000 | 13.1417 | 6.66614 |
| 110000 | 19.671 | 9.88171 |
| 130000 | 27.7176 | 13.8636 |



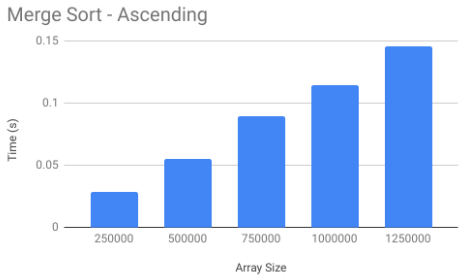

Analysis of the curves appears to indicate that the ascending case has behavior of O(n), while the descending and random cases have behavior of $O(n^2)$.

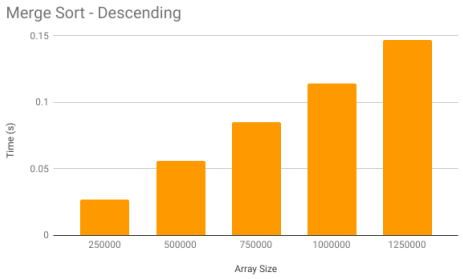
Based on this assumption, the time to sort an an array of size 10,000,000 can be estimated by solving the ratio $t_1/t_2 = O_1/O_2$:

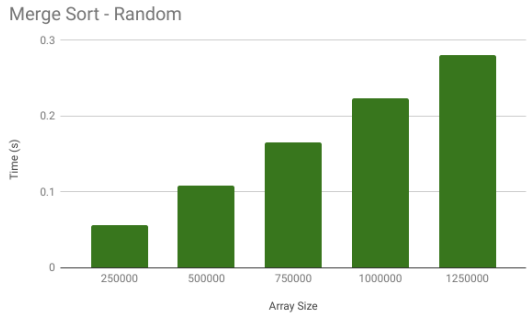
| Size | Ascending (s) | Descending (s) | Random (s) |
|---|---|---|---|
| 10000000 | .03732 | 165750 | 84383 |

**Merge Sort**

| Size | Ascending (s) | Descending (s) | Random (s) |
|---|---|---|---|
| 250000 | 0.028821 | 0.026881 | 0.056024 |
| 500000 | 0.055499 | 0.05605 | 0.107812 |
| 750000 | 0.089223 | 0.084968 | 0.165064 |
| 1000000 | 0.114636 | 0.114441 | 0.223062 |
| 1250000 | 0.146109 | 0.146994 | 0.281056 |



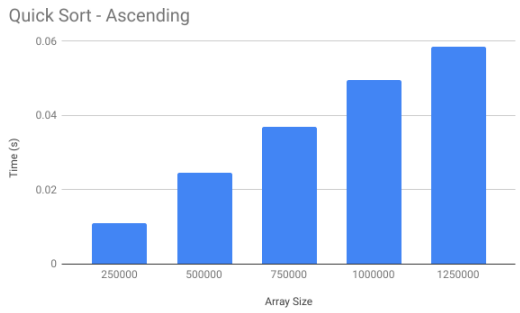

Analysis of the data appears to indicate that all three cases have behavior of O(n*log(n)).

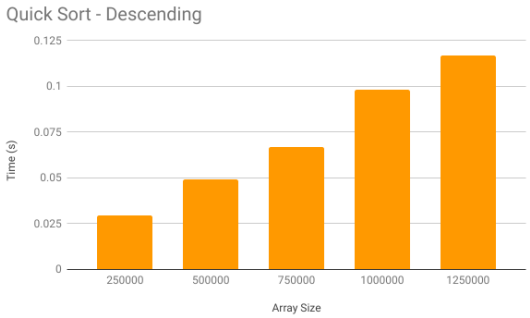
Based on this assumption, the time to sort an an array of size 10,000,000 can be estimated by solving the ratio $t_1/t_2 = O_1/O_2$:

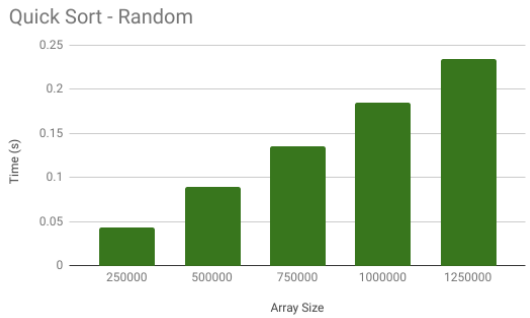
| Size | Ascending (s) | Descending (s) | Random (s) |
|---|---|---|---|
| 10000000 | 1.33742 | 1.335145 | 2.60239 |

**Quick Sort**

| Size | Ascending (s) | Descending (s) | Random (s) |
|---|---|---|---|
| 250000 | 0.010942 | 0.029608 | 0.043318 |
| 500000 | 0.0245 | 0.049031 | 0.089779 |
| 750000 | 0.036879 | 0.066811 | 0.135556 |
| 1000000 | 0.049661 | 0.098361 | 0.184359 |
| 1250000 | 0.058512 | 0.117093 | 0.233724 |




Analysis of the data appears to indicate that all three cases demonstrate O(n*log(n)) behavior.

Based on this assumption, the time to sort an an array of size 10,000,000 can be estimated by solving the ratio $t_1/t_2 = O_1/O_2$:

| Size | Ascending (s) | Descending (s) | Random (s) |
|---|---|---|---|
| 10000000 | 0.57938 | 1.1475 | 2.15086 |