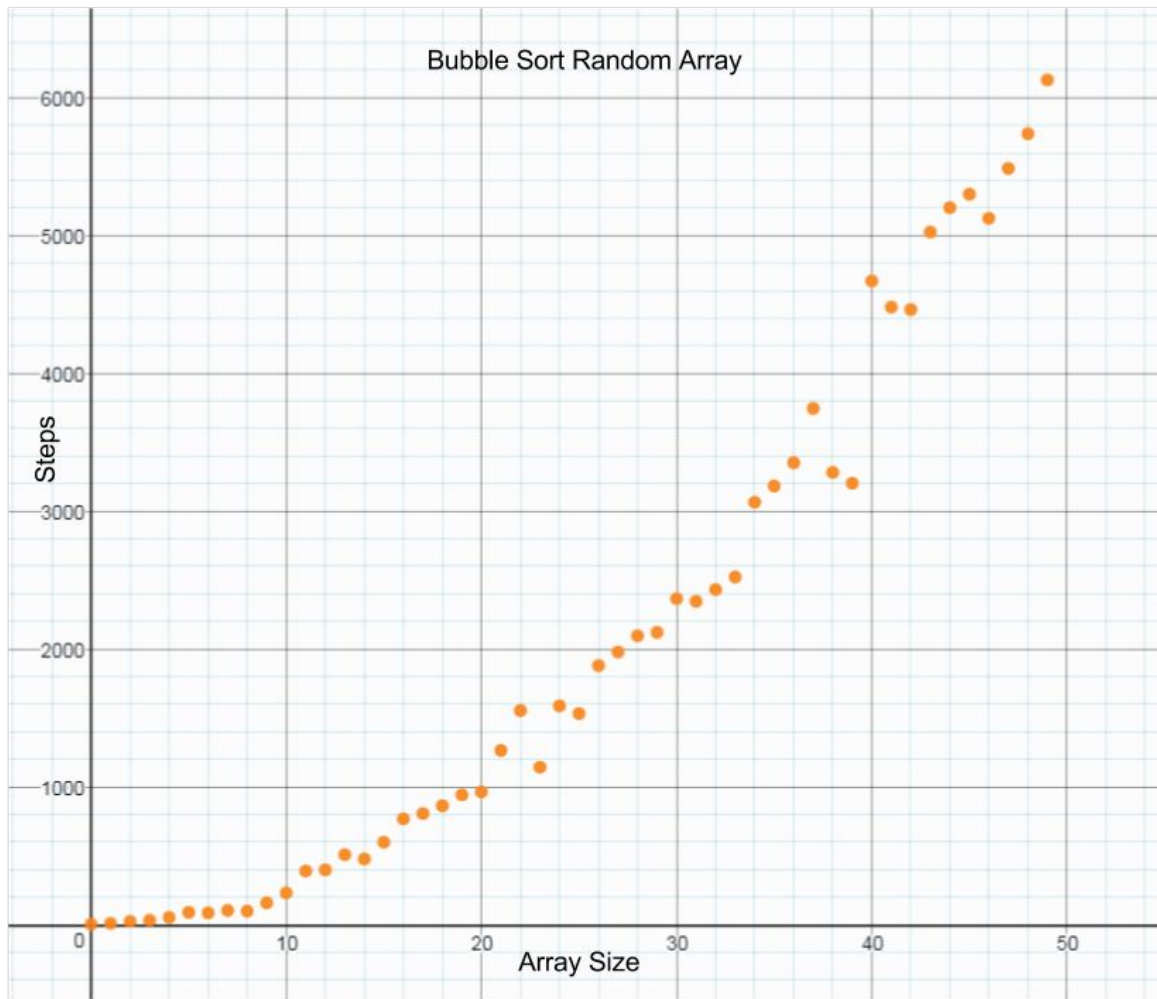
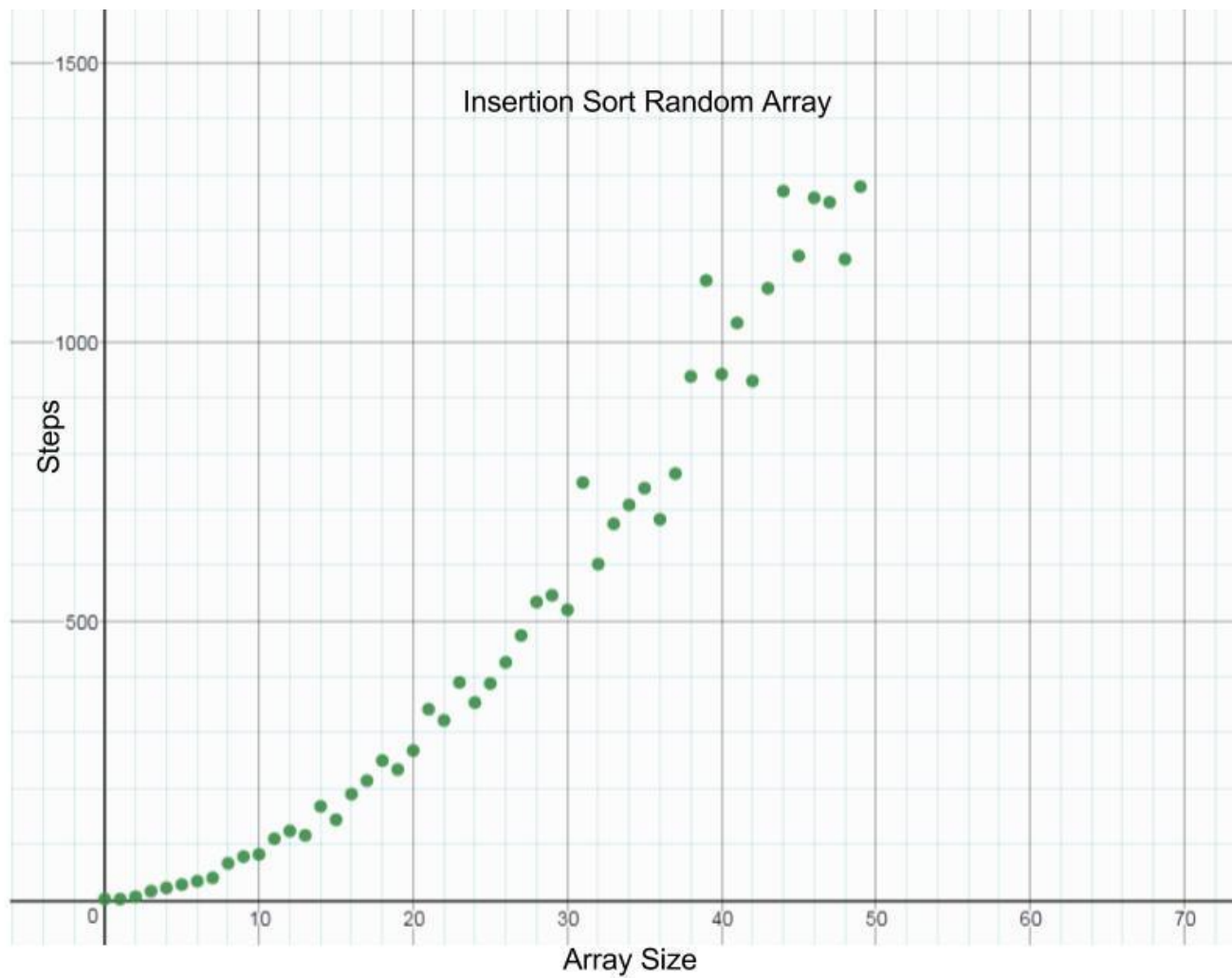
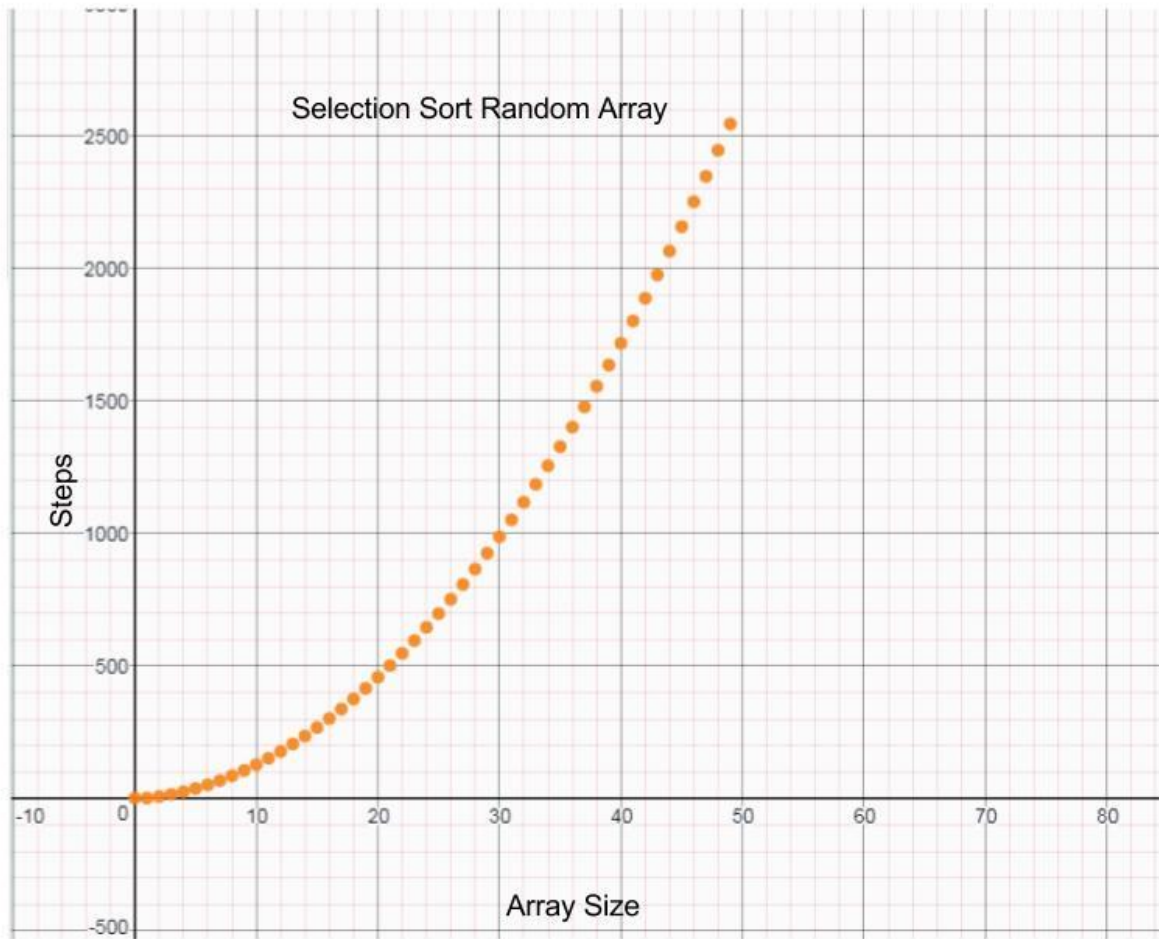

Quadratic Sorts

— Mayaank Vadlamani & Kashif
Peshimam —

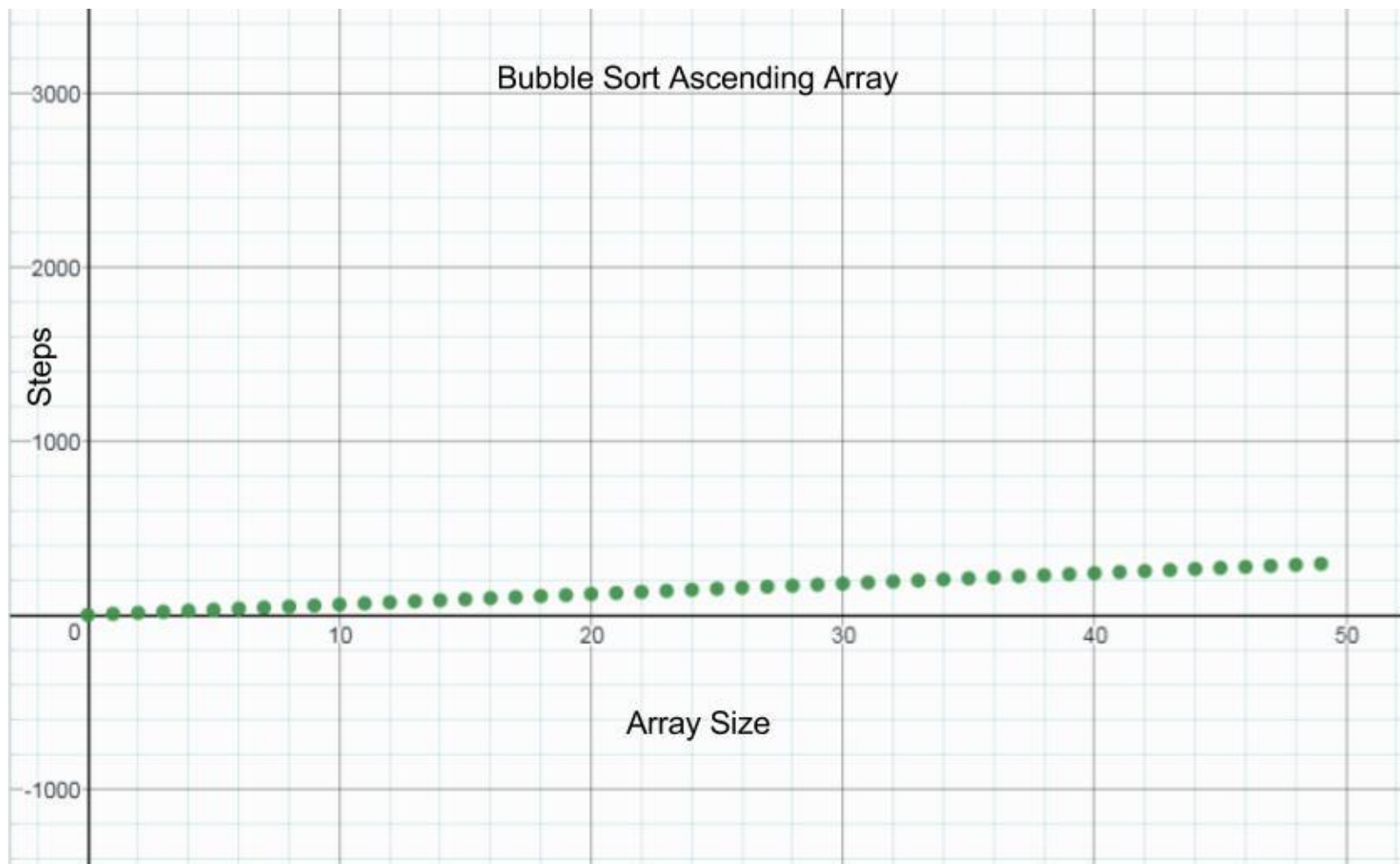
Random



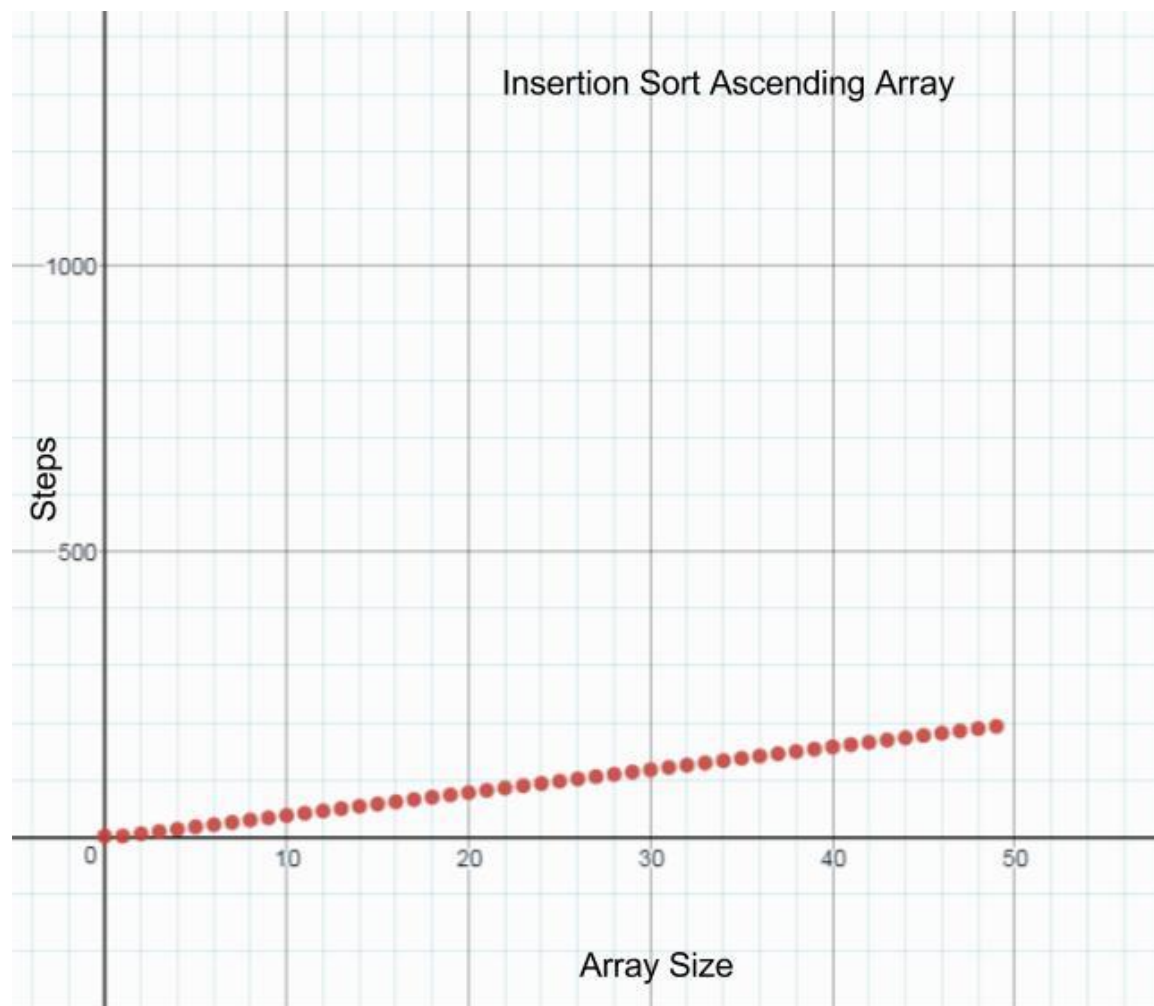


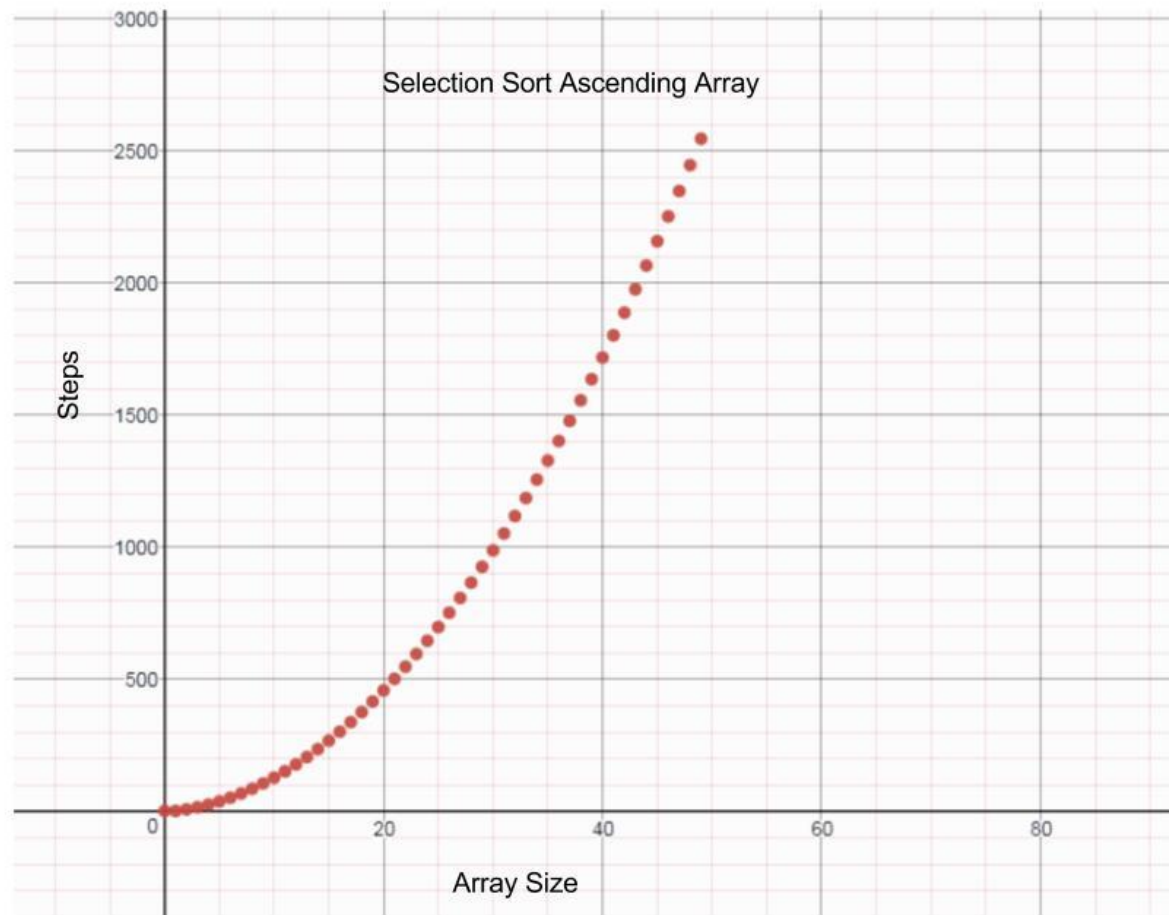


Ascending



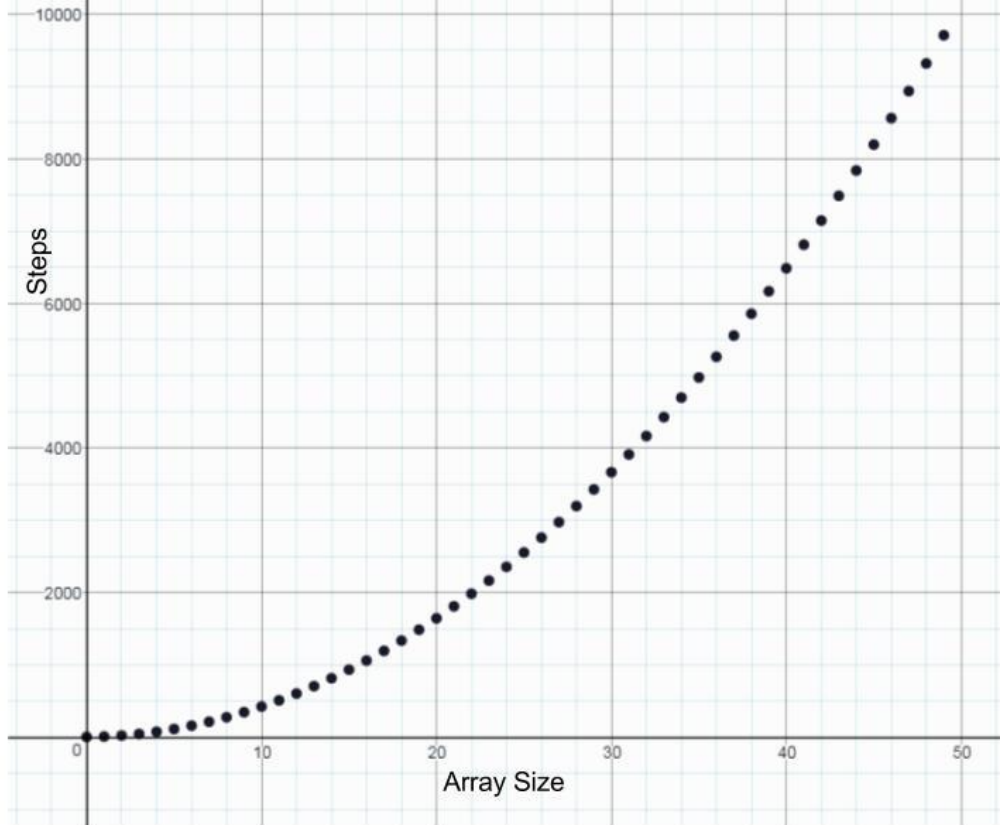
Insertion Sort Ascending Array

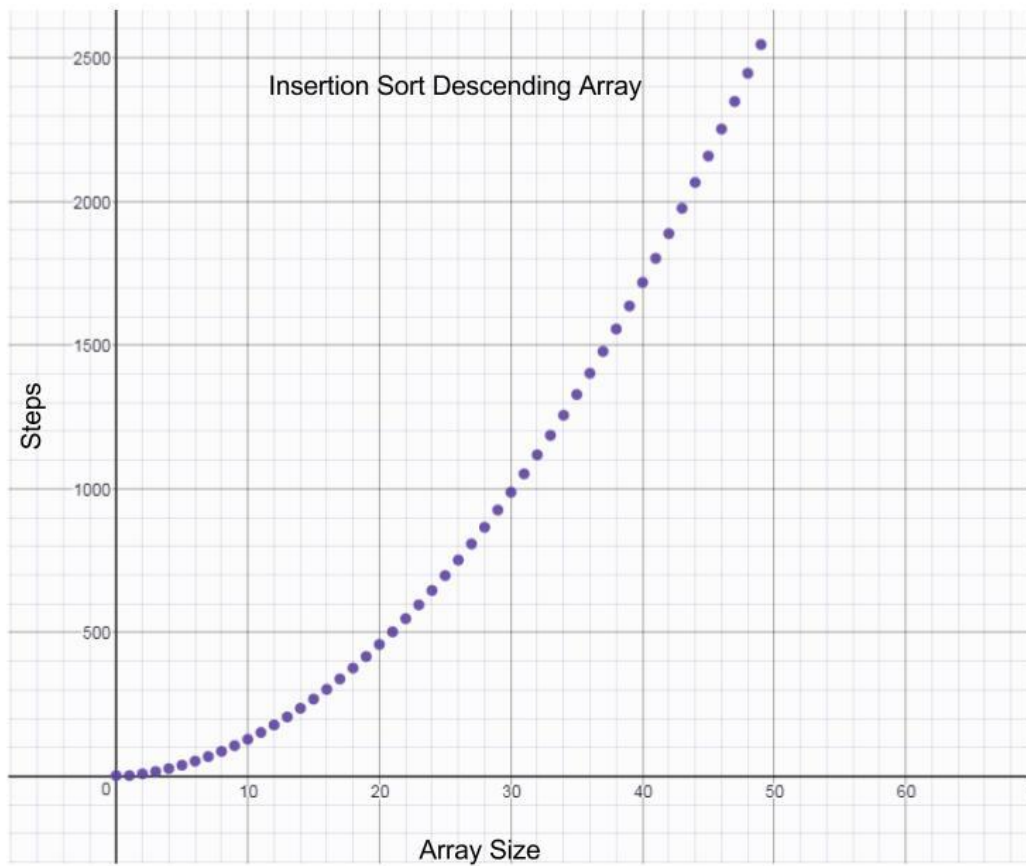


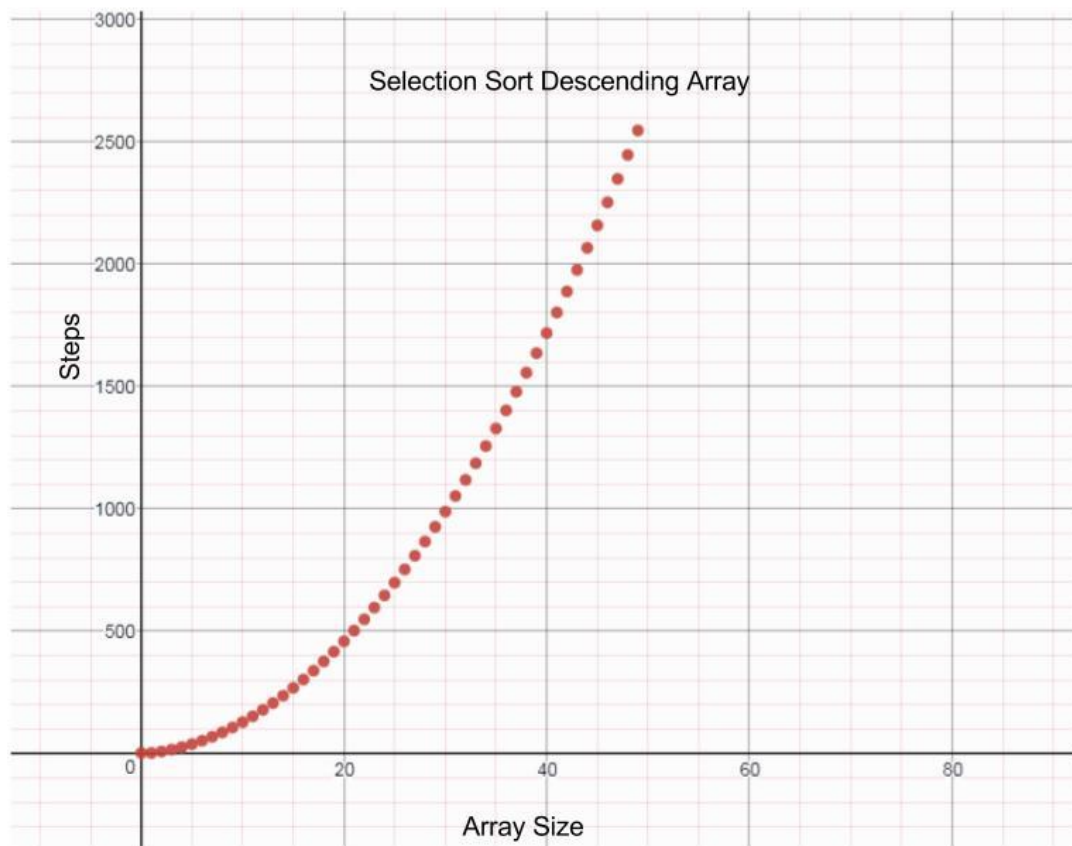


Descending

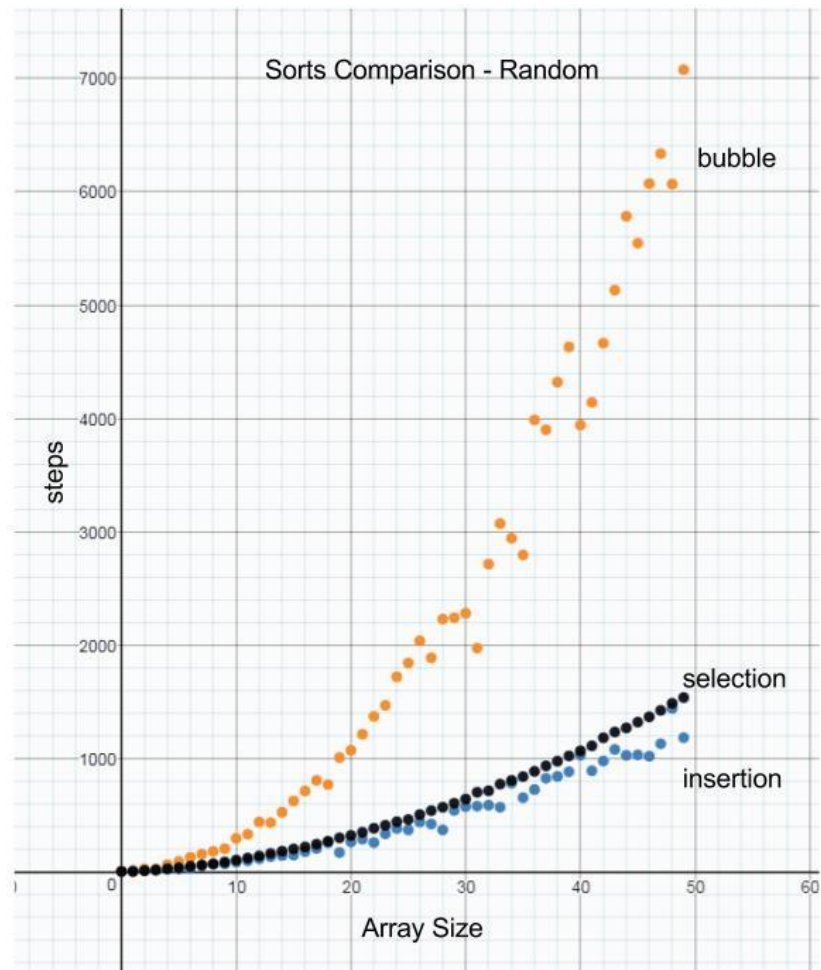
Bubble Sort Descending Array







All





Explain why these are called quadratic sorts. $O(n^2)$

- Number of comparisons increases as a quadratic relationship with array size
- Ex: Insertion sort
 - Outer loop runs N times
 - Inner loop runs $N/2$ times
 - $N * N/2 = N^2$

Which is the most efficient sort of a random array? Why?

- Insertion Sort
 - Insertion sort provides a $O(n^2)$ worst case algorithm that adapts to $O(n)$ time if data is nearly sorted.
 - requires less memory
 - Fewer comparisons

Which is the least efficient sort of a reverse ordered array? Why?

- Bubble Sort
 - Because of a large amount of possible swaps.
 - It compares every element to a lot of other elements. That slows it down
 - Worst case: $O(n^2)$
 - Average Case: $O(n^2)$
 - Best Case: $O(n)$

Which of these sort situations will produce a linear relationship $O(n)$. Why?

- Bubble and Insertion Sort with ascending ordered array
 - The array is already sorted
 - So the methods only pass through once ($O(n)$)
 - do not swap anything, everything is already in order.