Random Array

Elizabeth Goodwin

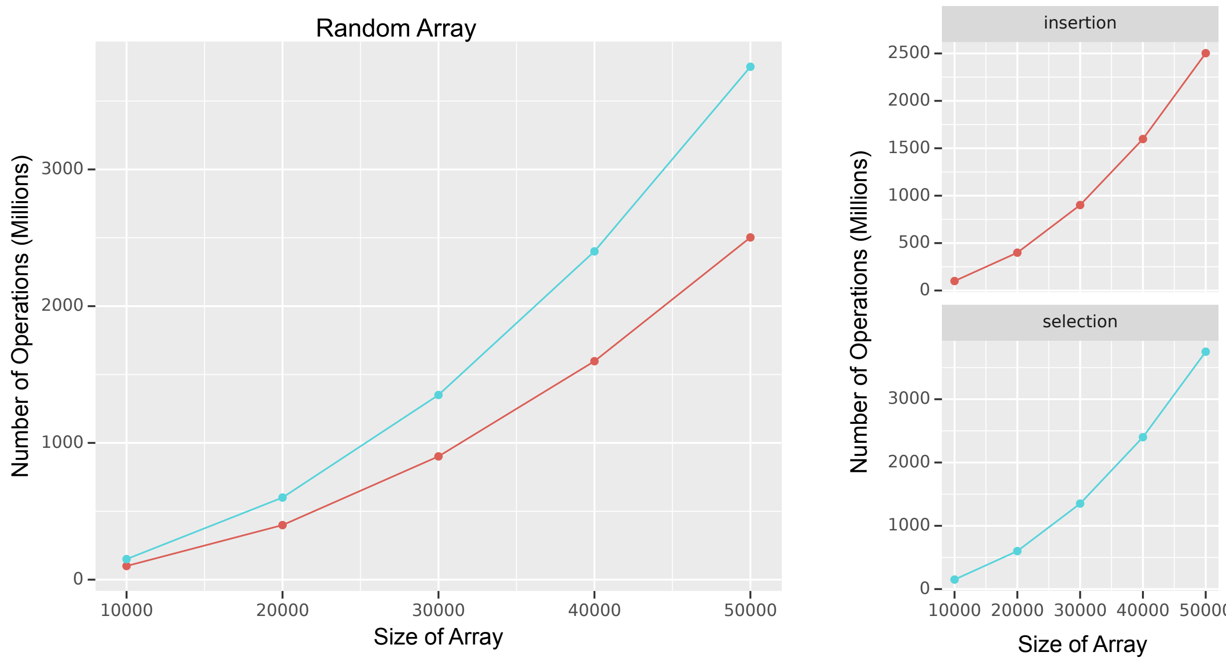

Figure 1: Random Array

Random presented pretty interesting results, with insertion sort winning over selection sort, but only by a limited amount. This a middle case between having the array be in order and having it be in reverse order, and should be the most realistic test. Selection sort as an algorithm is pretty consistent between ordering patterns. For each element of the array to sort, it has to loop over every not already sorted element and find the lowest element possible. As there is no way to know it is the lowest element without scanning all elements, it must scan all the unsorted elements no matter what.

Insertion sort the elements as it reads them. That is, it compares the new element to all of it's existing, already sorted elements. As those elements are already sorted and in order, it just moves left element by element, checking the one next to itself, and swapping them until it finds one less than itself. Once that happens, it knows that it has moved it into the correct order. This process continues for all in the set. The fewer swaps insertion sort has to do before inserting in the correct position, the more efficient it will be. Random sort is neither the worst (decreasing) or the best (increasing), but it not being the worst theoretical possible means it does not have to check as many elements as selection sort.

Insertion sort does require more operations to do each check, however, meaning a sufficiently close to decreasing order array will eventually be outdone by selection sort. This does not appear to be the case here, making Insertion sort the clear winner.