The example algorithm implements a quick sort. The quick sort is an example of a divide and conquer algorithm. In a divide and conquer algorithm, the array of values to be sorted is segmented into sub-arrays which are then further divided into sub-arrays. In the quicksort this is accomplished by selecting what is called a pivot value. The selection of the pivot value is somewhat arbitrary. In this algorithm we simply select left and right values and whatever happens to be in those positions within the array are used to find the mid point or the pivot value. The array is essentially divided along the pivot value. All values smaller than the pivot value are moved to the left array. All values larger than the pivot are moved to the right array resulting in two arrays. The two arrays are then recursively sorted. The pivot value can have a lot of impact on the efficiency of the algorithm. In the average case the complexity of the quicksort is O(n log n). The larger the array the more the algorithm tends toward the average case. In the worst case it may have O(n^2) complexity. It is for this reason that the quicksort is typically used when sorting arrays with a large n.

The number of exchanges required to sort the array using the insertion sort was 114. For the quicksort with its arbitrarily selected pivot point the number of required exchanges was 28 which is a bit more than O(n) which is consistent with the average case of O(n log n).