* Insertion sort is a sorting algorithm with a quadratic running time. Mostly used to sort smaller arrays.

Insertion sort is mostly used for arrays of smaller sizes.

* 1. Efficiency of sort
     1. Best case, if array is already sorted, insertion sort has a linear running time of O(n)
     2. Worst case, if array is sorted in reverse order, and average case, are both quadratic w/ a time of O(n^2)
     3. This makes insertion sort inefficient for large arrays but  is faster and better for smaller arrays than the divide-and-conquer sorts (merge and quick)
  2. Advantages
     + 1. Simple, short, easy to understand code
       2. Most efficient quadratic algorithm (selection and bubble)
       3. Adaptive; most efficient sort for data already or mostly sorted
       4. It is an in-place algorithm, meaning it only requires a constant amount O(1) of additional memory space
       5. Can sort lists as they are received
  3. When you might want to use insertion sort
     + 1. When sorting a mostly sorted array
       2. When sorting an array while receiving the data
       3. Quickly sorting a small amount of data

1. **Quick tips/troubleshoot**

If you get an index out of bounds error, check where your for loop is running to as well as what index your loop is starting at. Other than that, insertion sort is fairly straightforward, so there is not much to elaborate on.