Understanding Insertion Sorting

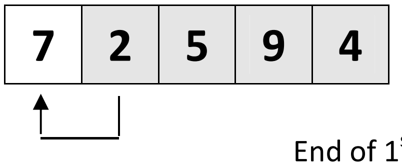
Insertion Sorting can be best understood if you pretend that you are playing a card game. If you pick up the cards you are dealt one at a time and order them in your hand one at a time … you are doing the exact Insertion Sorting process. When looking at data set (array of integers in this case), simply treat it as though you are picking up cards and organizing your cards one at a time. Here's an example …

Array

7	**2**	**5**	**9**	**4**

The 7 card is picked up and placed into my hand … it doesn't matter what the first card is, it does not need to be sorted. Technically, I don't need to do anything with the first card.

1st "Pass" -

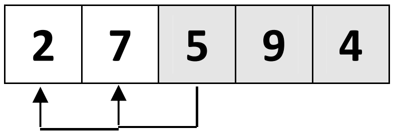


The 2 card is picked up and compared with the 7. A swap needs to be made, a swap is done.

End of 1st Pass →

2	**7**	**5**	**9**	**4**

2nd "Pass" -

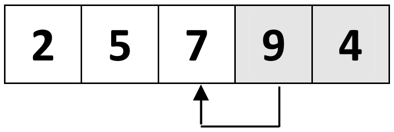


The 5 card is compared with the 7. The 5 is smaller, so another test is made with the 2. The 5 is placed in its proper position.

End of 2nd Pass →

2	**5**	**7**	**9**	**4**

3rd "Pass" -

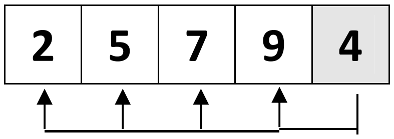


The 9 card is compared with the 7. No swap is needed so the process is stopped!

End of 3rd Pass →

2	**5**	**7**	**9**	**4**

4th "Pass" -



The 4 card is compared with the 9, which leads to a comparison with the 7 … and then the 5 … and then the 2. The home is finally found for the 4 and it is placed in its proper position.

End of 3rd Pass →

2	**4**	**5**	**7**	**9**

Congratulations, we've sorted our "hand"!

Read and understand the following Insertion Sorting method (envision picking up cards):

```
public static void sort(int theArray[ ])                    //method called sort can be called
{
    int numberToPlace;                                      //tempStorage for "card" we draw
    int n;                                                  //the "card in hand" we look at
    boolean keepGoing;                                      //should we keep looking?

    for(int i=1 ; i<theArray.length ; i++)                  //i=1 since 1st card we pick, known
    {                                                       //as theArray[0], needs no "order"
        numberToPlace = theArray[ i ];                      // "pick up a card" – next card really
         n = i – 1;                                         // n=index before the card you drew
        keepGoing = true;                                   // Make sure we keep going
         while( keepGoing && (n >= 0))                      // if we get to 1st card, don't loop
        {
            if(numberToPlace < theArray[n])                 //Card picked < last card in hand???
            {                                               // If yes …
                theArray[n+1] = theArray[n];                //Make room-move last card in hand
                n--;                                        //Move left a card in your hand
                if(n == -1)                                 //If there isn't a card to the left,
                    theArray[0] = numberToPlace;            //Make your new card the first card
            }                                               //While loop continues, check n--
            else                                            // If no …
            {
                keepGoing = false;                          // stop looping, don't check anymore
                theArray[n+1] = numberToPlace;              // put new card at the end
            }
        }
    }
}
```

Practice Insertion Sorting: What would each pass (stage) of insertion sorting do to (like pg 1):

Original Array:	4	7	5	3	9	1	6
Pass #1:	___	___	___	___	___	___	___
Pass #2:	___	___	___	___	___	___	___
Pass #3:	___	___	___	___	___	___	___
Pass #4:	___	___	___	___	___	___	___
Pass #5:	___	___	___	___	___	___	___
Pass #6:	___	___	___	___	___	___	___