**Insertion Sort**

Insertion sort is a simple sorting algorithm that works similar to the way you sort playing cards in your hands. The array is virtually split into a sorted and an unsorted part. Values from the unsorted part are picked and placed at the correct position in the sorted part.  
**Algorithm** –   
To sort an array of size n in ascending order:   
1: Iterate from arr[1] to arr[n] over the array.   
2: Compare the current element (key) to its predecessor.   
3: If the key element is smaller than its predecessor, compare it to the elements before. Move the greater elements one position up to make space for the swapped element.  
**Example:**



**Another Example:**  
**12**, 11, 13, 5, 6  
Let us loop for i = 1 (second element of the array) to 4 (last element of the array)  
i = 1. Since 11 is smaller than 12, move 12 and insert 11 before 12   
**11, 12**, 13, 5, 6  
i = 2. 13 will remain at its position as all elements in A[0..I-1] are smaller than 13   
**11, 12, 13**, 5, 6  
i = 3. 5 will move to the beginning and all other elements from 11 to 13 will move one position ahead of their current position.   
**5, 11, 12, 13**, 6  
i = 4. 6 will move to position after 5, and elements from 11 to 13 will move one position ahead of their current position.   
**5, 6, 11, 12, 13**  
 public class InsertionSort {  
 void sort(int[] arr) {  
 int n = arr.length;  
 for (int i = 1; i < n; i++) {  
 int key = arr[i];  
 int j = i - 1;  
  
 while (j >= 0 && arr[j] > key) {  
 arr[j + 1] = arr[j];  
 j = j - 1;  
 }  
 arr[j + 1] = key;  
 }  
 }  
  
 static void printArray(int arr[]) {  
 int n = arr.length;  
 for (int i = 0; i < n; ++i)  
 System.*out*.print(arr[i] + " ");  
  
 System.*out*.println();  
 }  
  
 public static void main(String args[]) {  
 int arr[] = {12, 11, 13, 5, 6};  
  
 InsertionSort ob = new InsertionSort();  
 ob.sort(arr);  
  
 *printArray*(arr);  
 }  
}