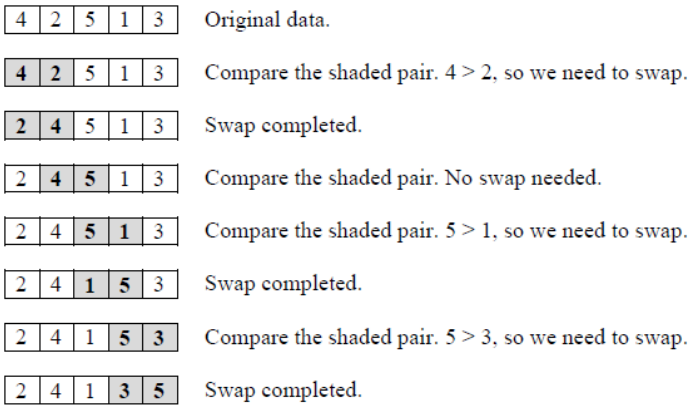
Bubble Sort (not part of the AP A test)

Best	Average	Worst
O(n)	$O(n^2)$	$O(n^2)$

The Bubble sort uses an **incremental** approach. The following shows the sequence of steps in a Bubble Sort:



After the first pass we notice that the largest value (5) has "bubbled" its way to the end of the list; however, the array is still not in order. Continue to repeat this process **until no swaps are made**. Only then is the list in order. On each subsequent pass the next largest value will "bubble" its way to its correct position near the end of the list.

Making the swap:

Before presenting a method that will perform a Bubble sort we need to first understand how to swap the contents of two variables. This procedure is at the heart of several types of sorting. Suppose we wish to interchange the value of integers *p* and *q* and that their original values are:

p = 5 and q = 79

When finished with the swap, their values will be:

p = 79 and q = 5

This is accomplished with the following code where you will notice the presence of *int temp* which serves as a safe haven for the first variable while the swap is being made.

```
temp = p;
p = q;
q = temp;
```

A Bubble Sort method:

The following method makes use of a similar swap where, instead, array values are used. You will also notice the use of a *boolean loopSomeMore* variable. Just under the sample sequence of steps above, the instruction, “Continue to repeat this process until no swaps are made.”, is implemented by usage of this *boolean* variable.

```
public static void sort(int a[]) //Bubble Sort
{
        boolean loopSomeMore;
        do
        {
                loopSomeMore = false;
                for(int j = 0; j < a.length -1; j++)
                {
                        if(a[j] > a[j+1])
                        {
                                //swap a[j] and a[j+1]
                                int temp = a[j];
                                a[j] = a[j+1];
                                a[j+1] = temp;

                                loopSomeMore = true;
                        }
                }
        }
        while(loopSomeMore);
}
```

Very, very slow:

This Bubble Sort is the slowest and most inefficient of all the sorting routines. It should only be used if you have a very few items to sort (say, 50 items or less). If you had, for example, 10,000 items to sort, this routine could literally take hours to run. **It is dreadfully slow**. So, why do we present it if it is so slow? Of all the sorting routines, it is also the **simplest to understand** and is therefore, a starting point for our study of sorting.