**Comb Sort:**

Comb Sort is mainly an improvement of Bubble Sort. Bubble sort always compares adjacent values. So all [inversions](https://www.geeksforgeeks.org/counting-inversions/) are removed one by one. Comb Sort improves on Bubble Sort by using gap of size more than 1. The gap starts with a large value and shrinks by a factor of 1.3 in every iteration until it reaches the value 1. Thus Comb Sort removes more than one [inversion counts](https://www.geeksforgeeks.org/counting-inversions/) in a single run.

The shrink factor has been empirically found to be 1.3 .

**Time Complexity:**

Worst case complexity of this algorithm is O (n2) and the Best Case complexity is O (n).

**Initializing:**

In the following code, we initialized an array in main and passing an array from main to the method for sorting where **n** and **gap** are initializedto array’s length.

Initialized Boolean **mark** to false.

**Working:**

Starting from while loop, which is working if **mark** equals to false and when **mark** gets true, our first iteration will complete.

Inside while loop first we find ‘Gap’.

**Gap = Gap / 1.3**

Here in this code we find gap by multiplying it by 10 and dividing it by 13 because gap is needed as integer value to run the code and dividing it by 1.3 will implicitly convert it into float.

**Gap = (Gap \* 10) / 1.3**

After getting the gap a condition checks if the gap value is less than 1 or not. If the value is less than 1 then condition sets the gap as one and also sets the **mark** as **true.**

After this condition a nested for loop which is runs up to n times where n (number of elements in array). This loop contains a condition which checks the numbers with a certain gap and swaps if needed and at lastly sets **mark** as false so that next iteration can run to check if it’s sorted.

This iteration will run until gap is equal to 1 and after the while loop ends and we will get a sorted array, which is returned to main.

**Using:**

In this code you will find one class, inside class there are two methods. One is main and the other is for comb sort.

Simply call the sort method in main passing one argument which is the array to be sorted.

The program runs and returns a sorted array back to main which can be used as like.