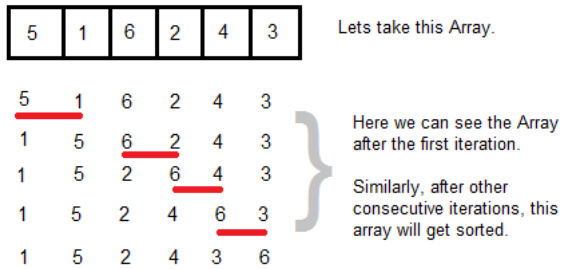
Bubble Sorting

Bubble Sort is an algorithm which is used to sort **N** elements that are given in a memory for eg: an Array with **N** number of elements. Bubble Sort compares all the element one by one and sort them based on their values.

It is called Bubble sort, because with each iteration the smaller element in the list bubbles up towards the first place, just like a water bubble rises up to the water surface.

Sorting takes place by stepping through all the data items one-by-one in pairs and comparing adjacent data items and swapping each pair that is out of order.



Sorting using Bubble Sort Algorithm

Let's consider an array with values {5, 1, 6, 2, 4, 3}

Coming Soon

```
int a[6] = {5, 1, 6, 2, 4, 3};
int i, j, temp;
for(i=0; i<6, i++)
{
  for(j=0; j<6-i-1; j++)
  {
    if( a[j] > a[j+1])
    {
      temp = a[j];
      a[j] = a[j+1];
      a[j+1] = temp;
    }
  }
}
//now you can print the sorted array after this
```

Above is the algorithm, to sort an array using Bubble Sort. Although the above logic will sort and unsorted array, still the above algorithm isn't efficient and can be enhanced further. Because as per the above logic, the for loop will keep going for six iterations even if the array gets sorted after the second iteration.

Hence we can insert a flag and can keep checking whether swapping of elements is taking place or not. If no swapping is taking place that means the array is sorted and wew can jump out of the for loop.

```
int a[6] = {5, 1, 6, 2, 4, 3};
int i, j, temp;
for(i=0; i<6, i++)
{
  for(j=0; j<6-i-1; j++)
  {
    int flag = 0;        //taking a flag variable
    if( a[j] > a[j+1])
    {
      temp = a[j];
      a[j] = a[j+1];
      a[j+1] = temp;
      flag = 1;         //setting flag as 1, if swapping occurs
    }
  }
  if(!flag)             //breaking out of for loop if no swapping takes place
  {
    break;
  }
}
```

In the above code, if in a complete single cycle of j iteration(inner for loop), no swapping takes place, and flag remains 0, then we will break out of the for loops, because the array has already been sorted.

Complexity Analysis of Bubble Sorting

In Bubble Sort, n-1 comparisons will be done in 1st pass, n-2 in 2nd pass, n-3 in 3rd pass and so on. So the total number of comparisons will be

```
(n-1)+(n-2)+(n-3)+.....+3+2+1
Sum = n(n-1)/2
i.e O(n^2)
```

Hence the complexity of Bubble Sort is **O(n^2)**.

The main advantage of Bubble Sort is the simplicity of the algorithm.Space complexity for Bubble Sort is **O(1)**, because only single additional memory space is required for **temp** variable

Best-case Time Complexity will be **O(n)**, it is when the list is already sorted.

← Prev (introduction-to-sorting)

Next → (insertion-sorting)