

Reference: GeeksforGeeks

This is the simple and optimized code for **bubble sort**.

The worst time complexity is: $O(n^2)$.

Best time complexity is $O(n)$ when the array is already sorted.

Average time complexity $O\left(\frac{n(n+1)}{2}\right)$.

```
1  \\Function for bubble sort(Optimized version)
2  void bubble_sort(int arr[], int &size)
3  {
4      int i, j;
5      bool swapped;
6      for(i=0; i<size-1; i++)
7      {
8          swapped = false;
9          for(j=0; j<size-i-1; j++)
10         {
11             if(arr[j]>arr[j+1])
12             {
13                 swap(arr[j], arr[j+1]);
14                 swapped = true;
15             }
16         }
17         if(swapped == false)
18         {
19             break;
20             /*If no two elements were swapped, means the array
21             is already sorted. This method will save much more
22             time. Chill!*/
23         }
24     }
25 }
26
```

Listing 1: Bubble sort

Bubble sort is not practical sorting algorithm, cause it is more complex than other sorting algorithm. Bubble sort should be avoided in the case of large collection. It will not be efficient in the case of a reverse-ordered collection which will occur the time-complexity $O(n^2)$. If the smallest element is at the end of the list, it will take $n - 1$ passes to move it to the beginning.

Questions related to Sorting

1.Sort by name with integer ID.