

# Bubble Sorting

Bubble sort is a basic algorithm for arranging elements in correct order.

Steps in bubble sort —

$a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5 \quad a_6 \quad \dots \quad a_n$

In this sorting we have to bubble out the largest element and placed it to its correct position.

Then again bubble out 2nd largest element from rest array and placed it to its correct position. And so-on.

S-1  $a_1$  is compared with  $a_2$  if  $a_1$  is greater then swap  $a_1$  and  $a_2$  otherwise bubble element is now  $a_2$  and so-on.

Example—

0 1 2 3 4

5 4 9 8 7

$n=5$

Comparisons( $C$ ) = ?

Swap( $S$ ) = ?

Step 1 1. bubble element is 5

5 is compared with 4, 5 is greater then swap, now

4 5 9 8 7

$C=1$

$S=1$

2. bubble element is 5

5 is compared with 9, 5 is smaller then 9 no swapping occur.  $C=2$

3. bubble element is 9

9 is compared with 8, 9 is greater

$C=3$



Swap, now

 $S=2$ 

4 5 8 9 7

4. bubble element is 9

9 is compared with 7, 9 is greater  
swap, $C=4$  $S=3$ 

4 5 8 7 9

\* Now the biggest number 9 is at its correct position and our first pass is completed

Step 2 4 5 8 7 9

1 bubble element is 4

 $4 < 5$ , no swap $C=5$ 

4 5 8 7 9

2. bubble element is 5

 $5 < 8$ , no swap $C=6$ 

4 5 8 7 9

3. bubble element is 8

 $8 > 7$ , swap $C=7$ 

4 5 7 8 9

 $S=4$ 

\* Now the 2nd largest element is at its correct position.

Step 3 1. bubble element = 4

 $4 < 5$ , no swap $C=8$ 

2. bubble element = 5

 $5 < 7$ , no swap $C=9$







## Algorithm -

```

TC = 0, TS = 0
for (int i = 1; i < n; i++)
{
    n_of_C, swap = 0
    for (int j = 0; j < n - i; j++)
    {
        if (a[j] > a[j+1])
        {
            swap(a[j], a[j+1])
            swap++
            TS++
        }
        n_of_C++
        TC++
    }
    // print after Passes
    // if swap == 0 break
}
    
```

## Dry Run for Worst case -

5 4 3 2 1 (n=5)

- i = 1, i < 5 (True)
  - j = 0, j < 4 (True)
    - if (a[0] > a[1])
      - 5 > 4 (True)
        - swap (4 5 3 2 1)
        - swap = 1, TS = 1
    - C = 1, TC = 1
    - j = 1, j < 4 (True)
      - a[1] > a[2]
        - 5 > 3 (True)
          - swap (4 3 5 2 1)
          - swap = 2, TS = 2
      - C = 2, TS = 2



•  $j=2$  ,  $j < 4$  (True)  
 $a[2] > a[3]$   
 $5 > 2$  (True)  
 swap (1 3 2 5 1)  
 swap = 3 , TS = 3  
 C = 3 , TC = 3

•  $j=3$  ,  $j < 4$  (True)  
 $a[3] > a[4]$   
 $5 > 1$  (True)  
 swap (4 3 2 1 5)  
 swap = 4 , TS = 4  
 C = 4 , TC = 4

•  $j=4$  ,  $j < 4$  (false)  
 //print statements  
 Comparison in pass 1 = 4  
 Swapping " " " = 4  
 Array - 4 3 2 1 5

•  $i=2$  ,  $i < 5$  (True)  
 •  $j=0$  ,  $j < (5-2)[3]$  (True)  
 $a[0] > a[1]$   
 $4 > 3$  (True)  
 swap (3 4 2 1 5)  
 swap = 1 , TS = 5  
 C = 1 , TC = 5

•  $j=1$  ,  $j < 3$  (True)  
 $a[1] > a[2]$   
 $4 > 2$  (True)  
 swap (3 2 4 1 5)  
 swap = 2 , TS = 6  
 C = 2 , TC = 6



•  $j=2, j < 3$  (True)

$a[2] > a[3]$

$4 > 1$  (True)

Swap (3 2 1 4 5)

Swap = 3, TS = 7

C = 3, TC = 7

•  $j=3, j < 3$  (false)

// Print -

Comparisons in Pass 2 = 3

Swapping " " " = 3

Array - 3 2 1 4 5

•  $i=3, i < 5$  (True)

•  $j=0, j < 2$  (True)

$a[0] > a[1]$

$3 > 2$  (True)

Swap (2 3 1 4 5)

Swap = 1, TS = 8

C = 1, TC = 8

•  $j=1, j < 2$  (True)

$a[1] > a[2]$

$3 > 1$  (True)

Swap (2 1 3 4 5)

Swap = 2, TS = 9

C = 2, TC = 9

•  $j=2, j < 2$  (false)

•  ~~$i=4, i < 5$  (True)~~

// Print -

Comparison in Pass 3 = 2

Swapping " " " = 2

Array = 2 1 3 4 5



•  $i=4, i < 5$  (True)

•  $j=0, j < 1$  (True)

$a[0] > a[1]$

$2 > 1$  (True)

swap = (1 2 3 4 5)

swap = 1, TS = 10

C = 1, TC = 10

•  $j=1, j < 1$  (False)

// print -

Comparison in pass 4 = 1

Swapping " " = 1

Array = 1 2 3 4 5

•  $i=5, i < 5$  (False)

out of both loop.

Ans - TS = 10

TC = 10

Array = 1 2 3 4 5

TPass = 4

• Dry Run for Best Case -

1 2 3 4 5

•  $i=1, i < 5$  (True)

•  $j=0, j < 4$  (True)

$a[0] > a[1]$

$1 > 2$  (False)

C = 1, TC = 1

•  $j=1, j < 4$  (True)

$a[1] > a[2]$

$2 > 3$  (False)

C = 2, TC = 2



$$\begin{aligned} \bullet j=2, j < 4 \text{ (True)} \\ a[2] > a[3] \\ 3 > 4 \text{ (False)} \end{aligned}$$

$$C=3, TC=3$$

$$\begin{aligned} \bullet j=3, j < 4 \text{ (True)} \\ a[3] > a[4] \\ 4 > 5 \text{ (False)} \end{aligned}$$

$$C=4, TC=4$$

$$\bullet j=4, j < 4 \text{ (False)}$$

// Print -

After Print

$$\text{Swap} = 0$$

So break out of these loops.

$$T. \text{ Comparison} = 4$$

$$T. \text{ Swapping} = 0$$

$$\text{Array} = 1 \ 2 \ 3 \ 4 \ 5$$

\* Time Complexity

• In worst case -

$$\begin{aligned} \text{for } n=5, \text{ comparison in 1st Pass} &= 4 \quad (n-1) \\ \text{2nd " } &= 3 \quad (n-2) \\ \text{3rd " } &= 2 \quad (n-3) \\ \text{4th " } &= 1 \quad 1 \end{aligned}$$

$$\begin{aligned} \text{Sum of total comparison} &\rightarrow \\ 1 + 2 + 3 + \dots + (n-1) \end{aligned}$$

$$\Rightarrow \frac{(n-1)((n-1)+1)}{2}$$

$$\Rightarrow \frac{n(n-1)}{2} = \frac{n^2 - n}{2} = O(n^2)$$



• In best case-

For  $n=5$ , Comparison in 1st Pair = 4  
Swapping = 0

Sum of total comparisons -  
4 for  $n=5$

$$\Rightarrow n-1$$

$$\Rightarrow O(n)$$