# Bubble sort

→ ith round largest element → ith at right place

| i=0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 10 | 1 | 7 | 6 | 14 | 9 |

## Round 1

↓
1st largest element
↓
sort
↓
placed at end

5 comparison

1 10 7 6 14 9

1 7 10 6 14 9

1 7 6 10 14 9

1 7 6 10 14 9

1 7 6 10 9 | $\boxed{14}$ → right place

## Round 2

2nd largest element
↓
sort

4 comparison

1 7 6 10 9 | 14

1 6 7 10 9 | 14

1 6 7 10 9 | 14

1 6 7 9 | 10 | 14

## Round 3

3rd largest element
↓
sort

3 comparison

$1 \quad 6 \quad 7 \quad 9 \;\Big|\; 10 \quad 14$

$1 \quad 6 \quad 7 \quad 9 \;\Big|\; 10 \quad 14$

$1 \quad 6 \quad 7 \quad 9 \;\Big|\; 10 \quad 14$

$1 \quad 6 \quad 7 \quad \boxed{9} \;\Big|\; 10 \quad 14$

## Round 4

4th largest element
↓
sort

2 comparison

$1 \quad 6 \quad 7 \;\Big|\; 9 \quad 10 \quad 14$

$1 \quad 6 \quad 7 \;\Big|\; 9 \quad 10 \quad 14$

$1 \quad 6 \quad \boxed{7} \;\Big|\; 9 \quad 10 \quad 14$

## Round 5

5th largest element
↓
sort

1 comparison

$1 \quad 6 \;\Big|\; 7 \quad 9 \quad 10 \quad 14$

$1 \quad \boxed{6} \;\Big|\; 7 \quad 9 \quad 10 \quad 14$

## Round 6

→ single element left

↳ no need → already sorted in 5 rounds

$\boxed{n-1}$

total comparison = $5+4+3+2+1 \Rightarrow \frac{n(n-1)}{2}$

$O(n^2)$

Entire array is sorted

Condition

if $a < b$ ✗

$a > b$ (swap)

rounds = $n-1$

---

i

$0 \rightarrow < (n-1)$

$0 \rightarrow < (n-2)$

$0 \rightarrow < (n-3)$

⋮

u

$1 \rightarrow < (n-1)$

⋮

sorted

$O(n)$

Time complexity $\Rightarrow O(n^2)$

Space complexity $\Rightarrow O(1)$

---

optimisation

a b c d e $\Rightarrow$ no swap

$\boxed{a < b < c < d < e} \Rightarrow$ sorted

6n → 5 rounds

```
for ( int round = 1 ; round < n ; round++ )
{
   bool swaped = false ;
   for ( int j = 0 ; j < n - round ; j++ )
   {
         if ( arr[j] > arr [ j+1 ] )
         {
               swap ( arr ( j ] , arr ( j+1 ] )
               swaped = true ;
         } }
   if (swapped == false)
   {    break ; }
}
```

becoz we access j+1 element for comparison & round ↓ last ele sort in every round

| | | |
|---|---|---|
| Round 1 | → | j = 4 |
| Round 2 | → | j = 3 |
| Round 3 | → | j = 2 |
| Round 4 | → | j = 1 |
| Round 5 | → | j = 0 |

```
void bubbleSort(int arr[], int n)
   {
       for(int i = 0; i < n-1; i++){
           for(int j = i+1; j < n; j++){
               if(arr[j] < arr[i]){
                   swap(arr[i],arr[j]);
               }
           }
       }
   }
```