



# DESIGN AND ANALYSIS OF ALGORITHMS

## UE19CS251

---

**Shylaja S S**

Department of Computer Science  
& Engineering

# DESIGN AND ANALYSIS OF ALGORITHMS

---


## Bubble Sort

Major Slides Content: Anany Levitin

**Shylaja S S**

Department of Computer Science & Engineering

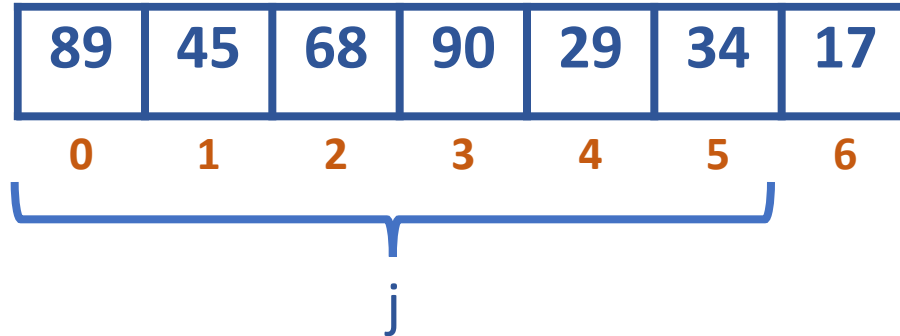
- Compare adjacent elements of the list and exchange them if they are out of order
- By doing it repeatedly, we end up bubbling the largest element to the last position on the list
- The next pass bubbles up the second largest element and so on and after  $n - 1$  passes, the list is sorted
- Pass  $i$  ( $0 \leq i \leq n - 2$ ) can be represented as follows:

$A[0], A[1], A[2], \dots, A[j] \quad ? \quad A[j+1], \dots, A[n-i-1] \mid A[n-i] \leq \dots \leq A[n-1]$   
  
 in their final positions

# DESIGN AND ANALYSIS OF ALGORITHMS

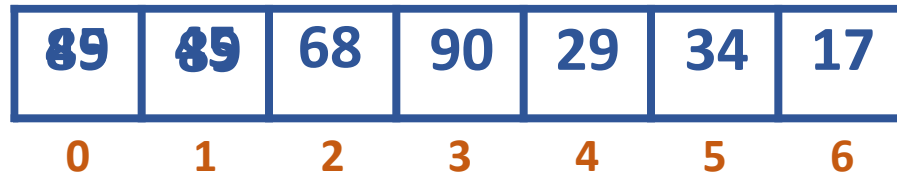
## Bubble Sort

$i = 0$



elements in  
final position

$j = 0$

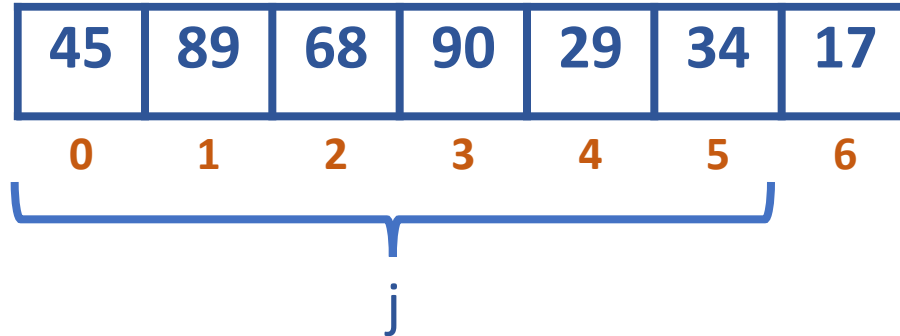


$45 < 89$  ☒ swap

# DESIGN AND ANALYSIS OF ALGORITHMS

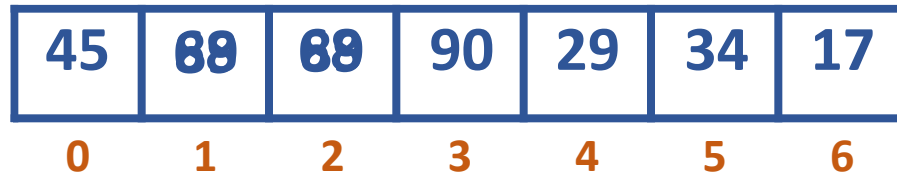
## Bubble Sort

$i = 0$



elements in  
final position

$j = 1$

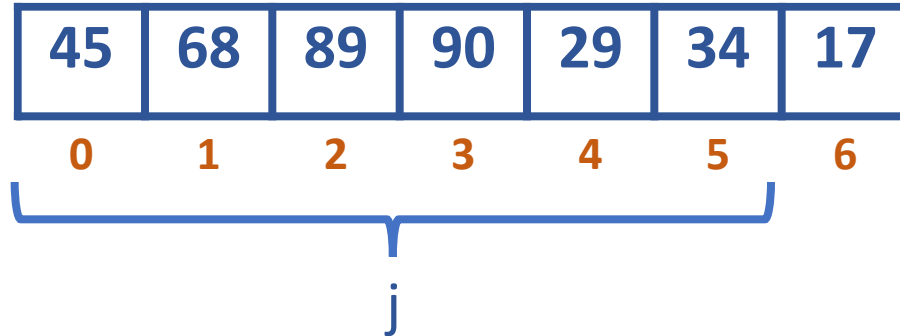


$68 < 89$  ☒ swap

# DESIGN AND ANALYSIS OF ALGORITHMS

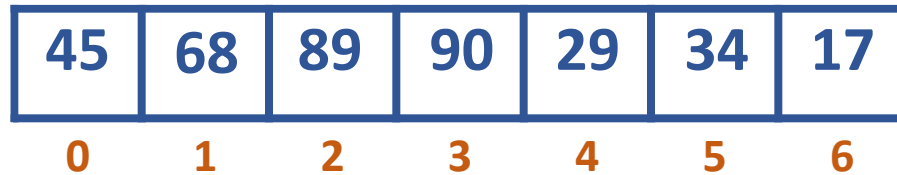
## Bubble Sort

$i = 0$



elements in  
final position

$j = 2$

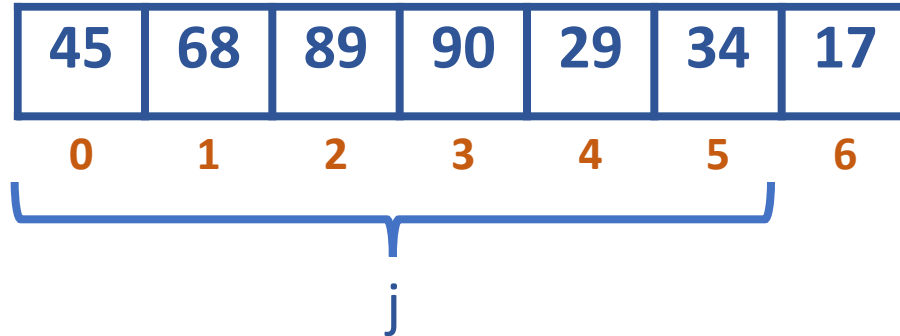


$90 < 89$   no swap

# DESIGN AND ANALYSIS OF ALGORITHMS

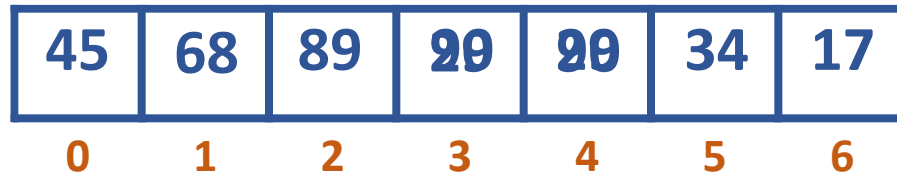
## Bubble Sort

$i = 0$



elements in  
final position

$j = 3$

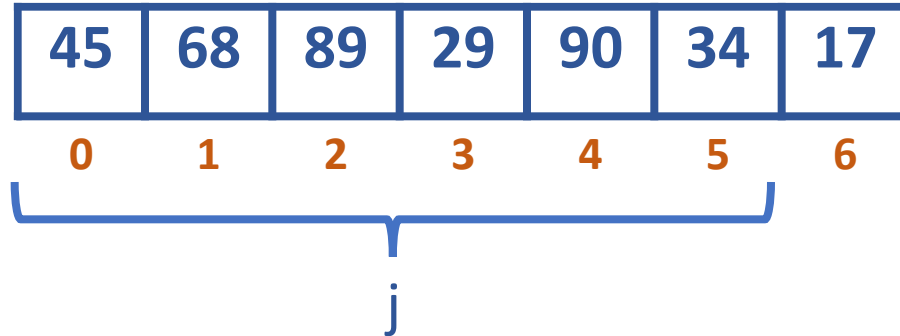


$29 < 90$  ☒ swap

# DESIGN AND ANALYSIS OF ALGORITHMS

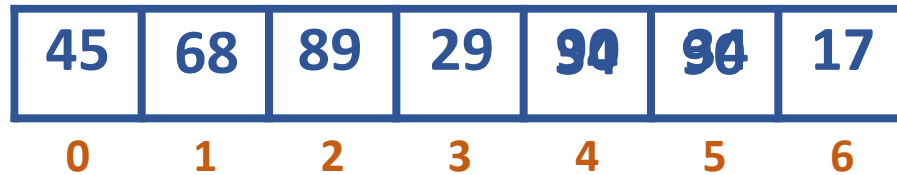
## Bubble Sort

$i = 0$



elements in  
final position

$j = 4$



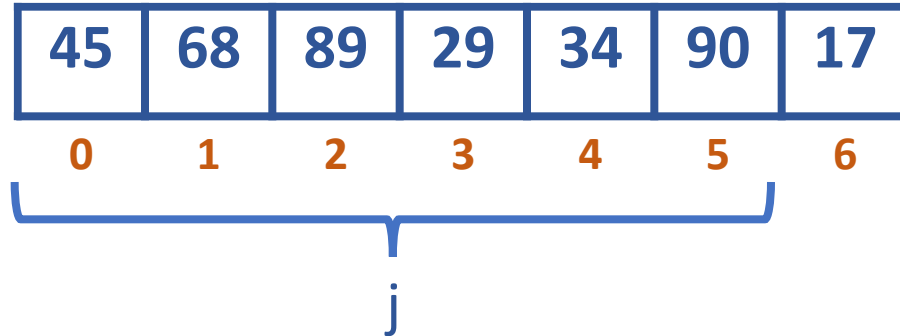
$34 < 90$  ☒ swap



# DESIGN AND ANALYSIS OF ALGORITHMS

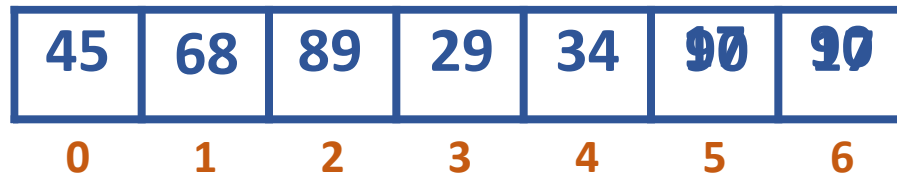
## Bubble Sort

$i = 0$



elements in  
final position

$j = 5$



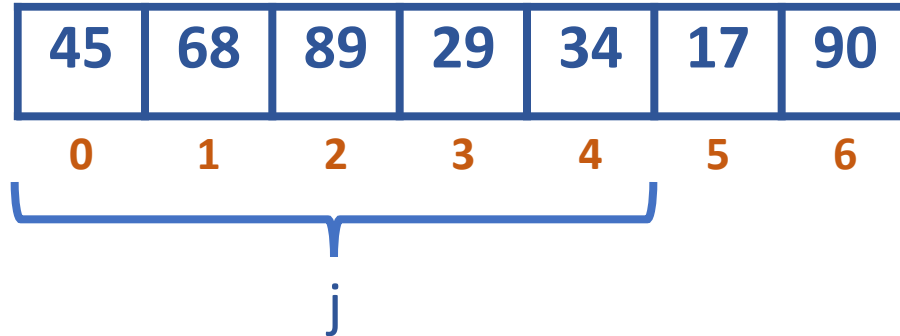
$17 < 90$  ☒ swap

# DESIGN AND ANALYSIS OF ALGORITHMS

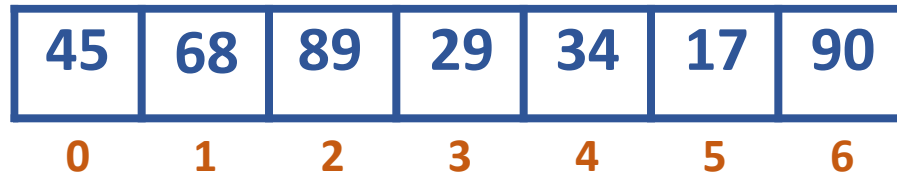
## Bubble Sort

After  
first  
iteration  
 $i = 1$

elements in  
final position



$j = 0$

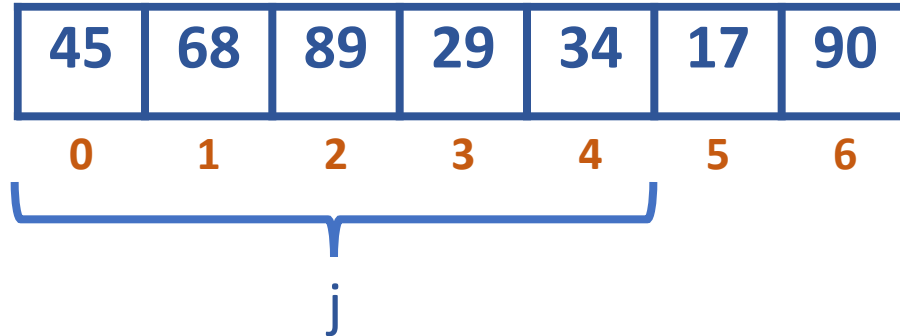


$68 < 45$   no swap

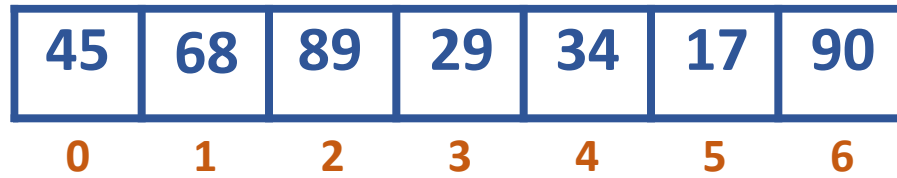
# DESIGN AND ANALYSIS OF ALGORITHMS

## Bubble Sort

$i = 1$



$j = 1$

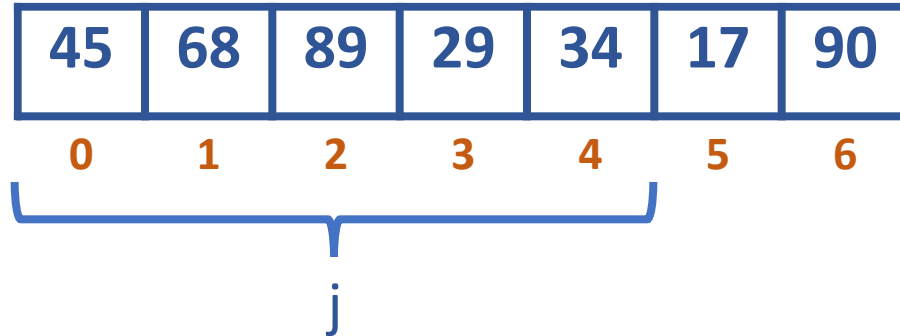


$89 < 68$   no swap

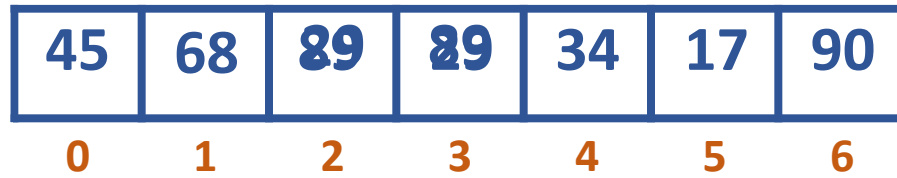
# DESIGN AND ANALYSIS OF ALGORITHMS

## Bubble Sort

$i = 1$



$j = 2$

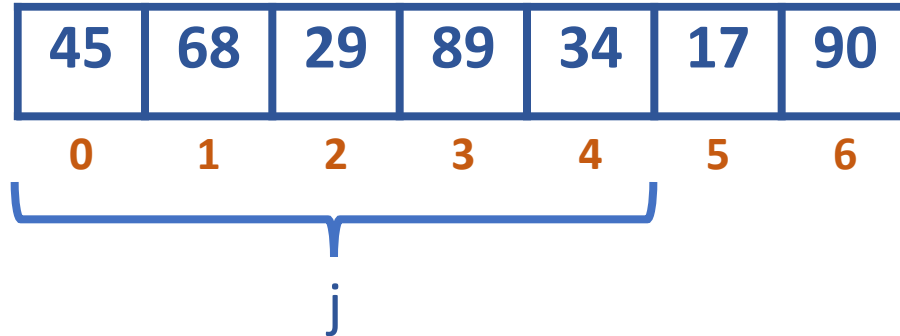


$29 < 89$  ☒ swap

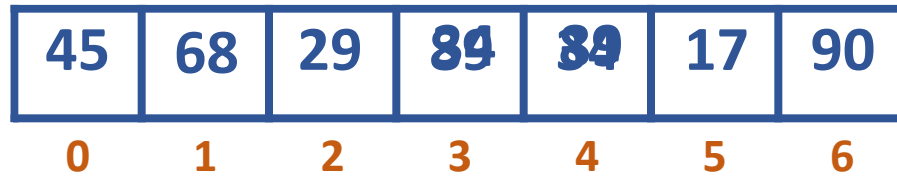
# DESIGN AND ANALYSIS OF ALGORITHMS

## Bubble Sort

$i = 1$



$j = 3$

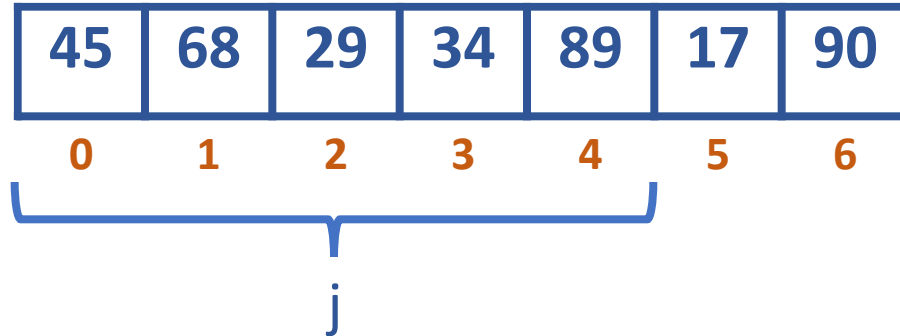


$34 < 89$  ☒ swap

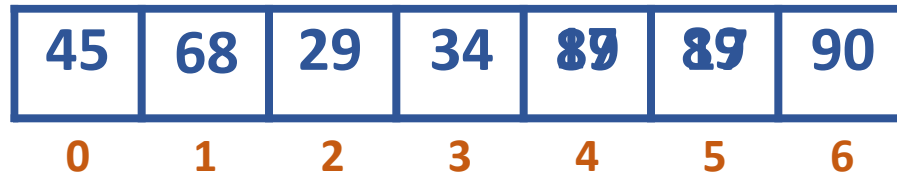
# DESIGN AND ANALYSIS OF ALGORITHMS

## Bubble Sort

$i = 1$



$j = 4$

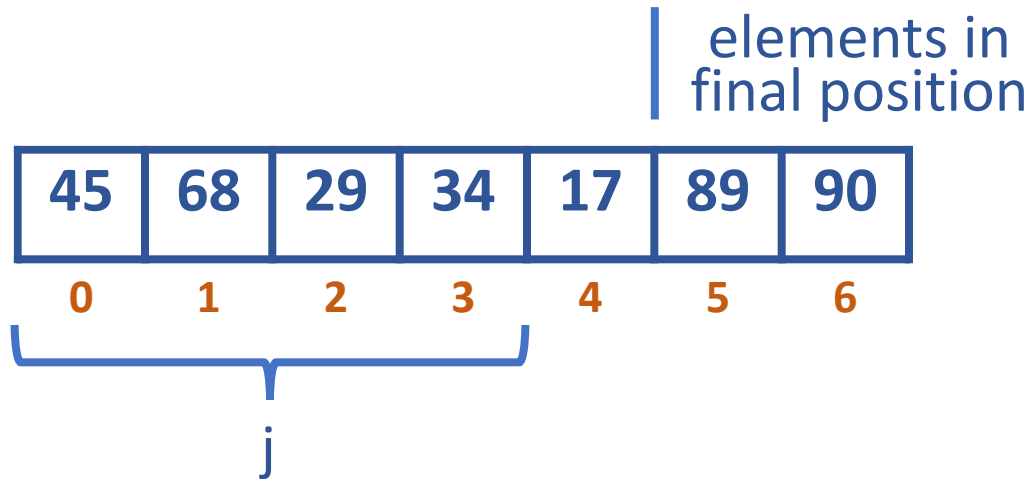


$17 < 89$  ☒ swap

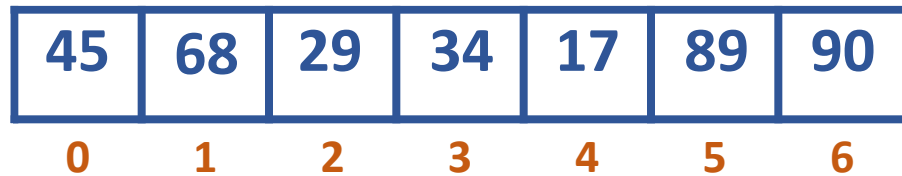
# DESIGN AND ANALYSIS OF ALGORITHMS

## Bubble Sort

After  
second  
iteration  
 $i = 2$



$j = 0$



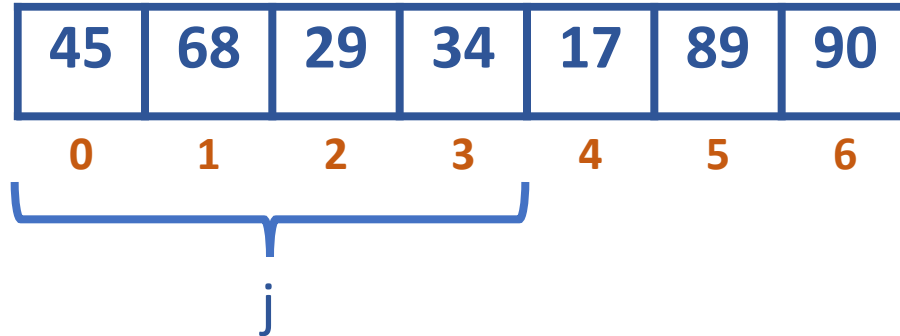
$68 < 45$   no swap

# DESIGN AND ANALYSIS OF ALGORITHMS

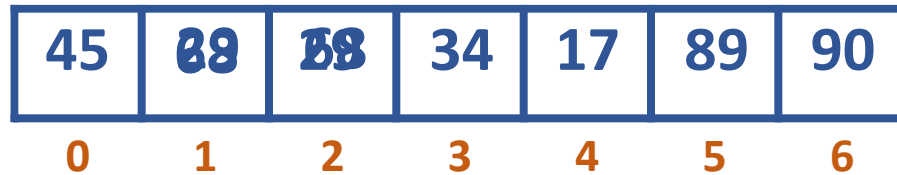
## Bubble Sort

elements in  
final position

$i = 2$



$j = 1$



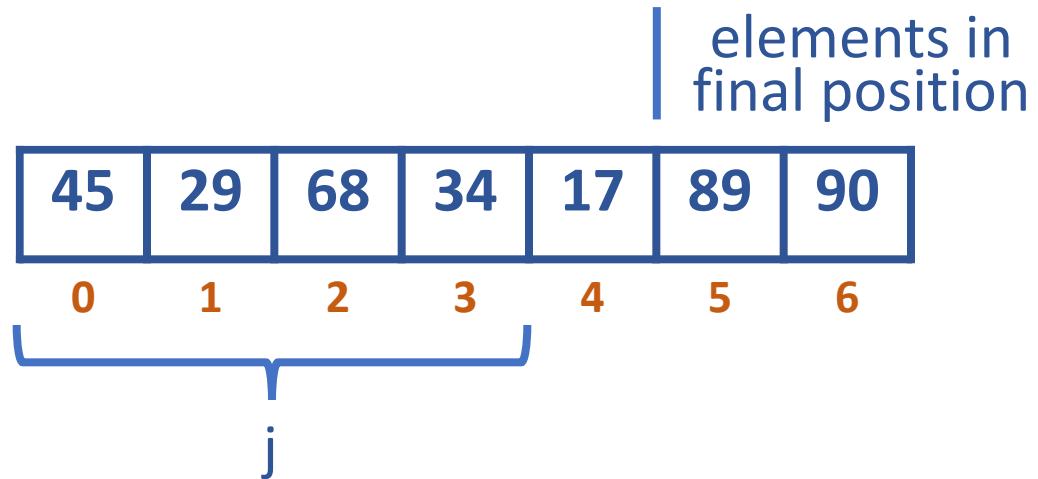
$29 < 68$  ☒ swap



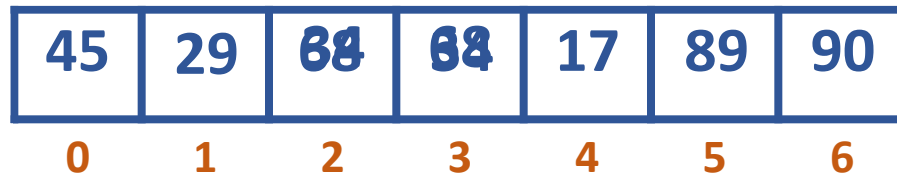
# DESIGN AND ANALYSIS OF ALGORITHMS

## Bubble Sort

$i = 2$



$j = 2$

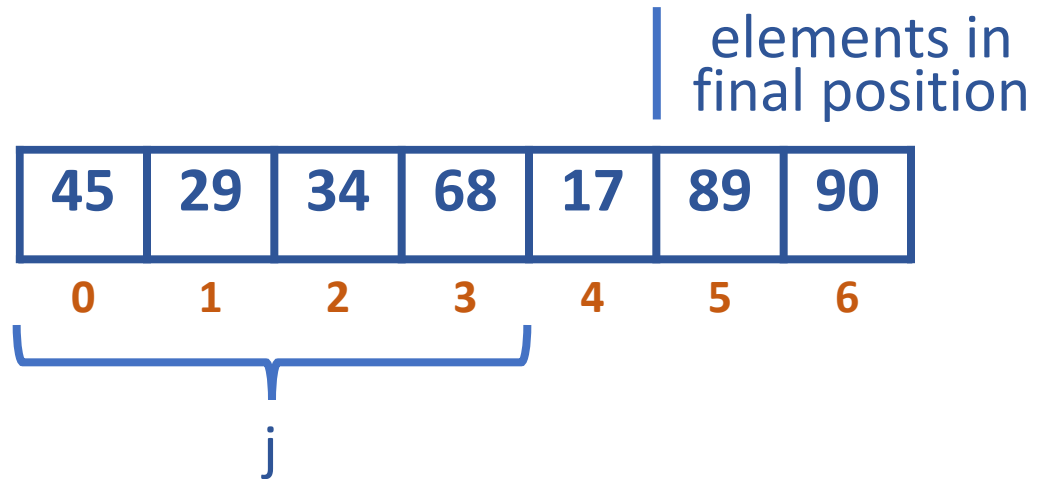


$34 < 68$  ☒ swap

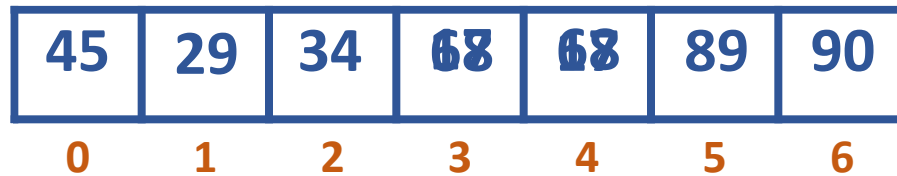
# DESIGN AND ANALYSIS OF ALGORITHMS

## Bubble Sort

$i = 2$



$j = 3$

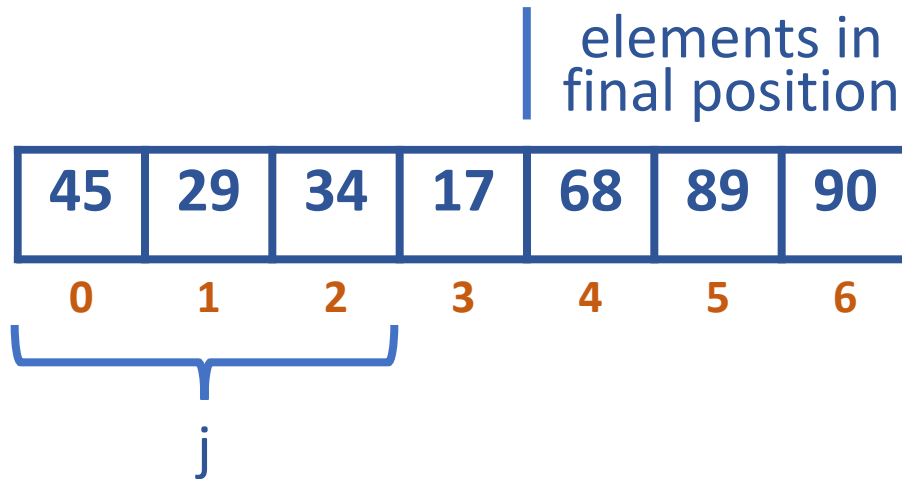


$17 < 68$  ☒ swap

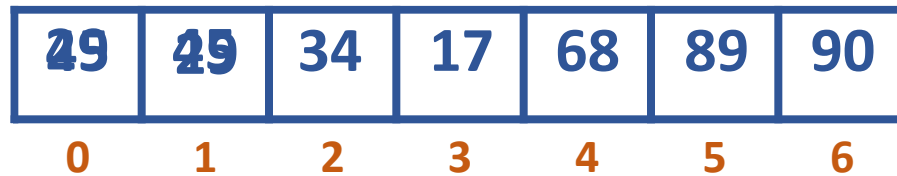
# DESIGN AND ANALYSIS OF ALGORITHMS

## Bubble Sort

After  
third  
iteration  
 $i = 3$



$j = 0$

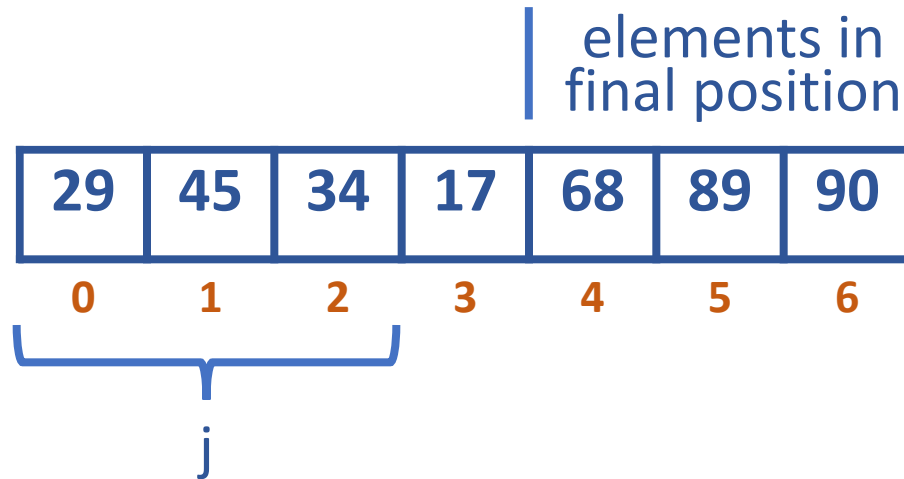


$29 < 45$  ☒ swap

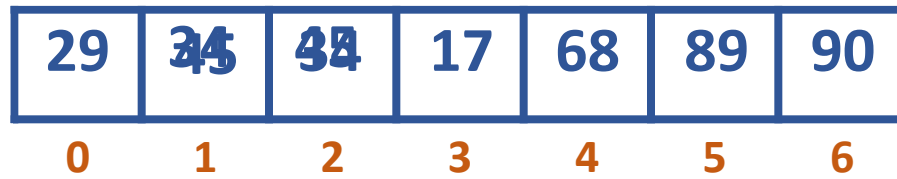
# DESIGN AND ANALYSIS OF ALGORITHMS

## Bubble Sort

$i = 3$



$j = 1$



$34 < 45$  ☒ swap

# DESIGN AND ANALYSIS OF ALGORITHMS

## Bubble Sort

elements in  
final position

$i = 3$

29	34	45	17	68	89	90
0	1	2	3	4	5	6

$j$

$j = 2$

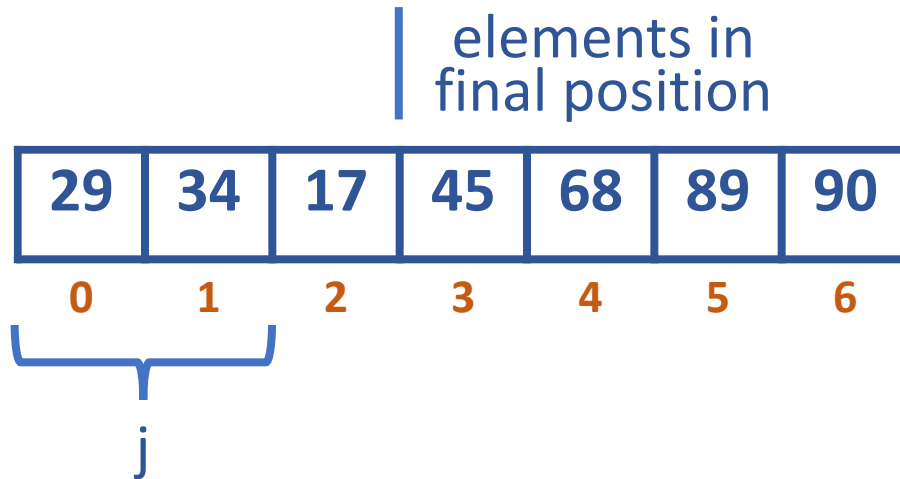
29	34	45	17	68	89	90
0	1	2	3	4	5	6

$17 < 45$  ☒ swap

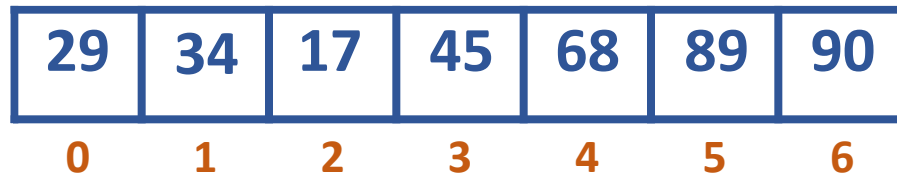
# DESIGN AND ANALYSIS OF ALGORITHMS

## Bubble Sort

After  
fourth  
iteration  
 $i = 4$



$j = 0$



$34 < 29$   no swap

# DESIGN AND ANALYSIS OF ALGORITHMS

## Bubble Sort

elements in  
final position

$i = 4$

29	34	17	45	68	89	90
0	1	2	3	4	5	6

$j$

$j = 1$

29	34	17	45	68	89	90
0	1	2	3	4	5	6

$17 < 34$  ☒ swap

# DESIGN AND ANALYSIS OF ALGORITHMS

## Bubble Sort

After  
fifth  
iteration  
 $i = 5$

elements in final position

29	17	34	45	68	89	90
0	1	2	3	4	5	6

j

$j = 0$

29	29	34	45	68	89	90
0	1	2	3	4	5	6

$17 < 29$  ☒ swap



# DESIGN AND ANALYSIS OF ALGORITHMS

## Bubble Sort

After  
sixth  
iteration

elements in  
final position

17	29	34	45	68	89	90
0	1	2	3	4	5	6

## Bubble Sort

---

ALGORITHM BubbleSort( $A[0 \dots n - 1]$ )

//Sorts a given array by bubble sort in their final positions

//Input: An array  $A[0 \dots n - 1]$  of orderable elements

//Output: Array  $A[0 \dots n - 1]$  sorted in ascending order

for  $i \leftarrow 0$  to  $n - 2$  do

    for  $j \leftarrow 0$  to  $n - 2 - i$  do

        if  $A[j + 1] < A[j]$  swap  $A[j]$  and  $A[j + 1]$

### Bubble Sort Analysis

$$\begin{aligned} C(n) &= \sum_{i=0}^{n-2} \sum_{j=0}^{n-2-i} 1 = \sum_{i=0}^{n-2} [(n-2-i) - 0 + 1] \\ &= \sum_{i=0}^{n-2} (n-1-i) = \frac{(n-1)n}{2} \in \Theta(n^2) \end{aligned}$$

Bubble Sort is a  $\Theta(n^2)$  algorithm



# THANK YOU

---

**Shylaja S S**

Department of Computer Science  
& Engineering

**[shylaja.sharath@pes.edu](mailto:shylaja.sharath@pes.edu)**