# COMP108
# Data Structures and Algorithms

## Bubble Sort Algorithm (Part I)

Professor Prudence Wong

pwong@liverpool.ac.uk

2022-23

## Outline

Bubble Sort algorithm

- using array
- using linked list

Learning outcome:

- Be able to describe and carry out asymptotic analysis of bubble sort algorithm

## Sorting

**Input:** a sequence of $n$ numbers $A_1, A_2, \cdots, A_n$

**Output:** arrange the $n$ numbers into **ascending order**, i.e., from smallest to largest

**E.g.:** If the input contains 5 numbers 132, 56, 43, 200, 10, then the output should be **10, 43, 56, 132, 200**.

There are many sorting algorithms:
bubble sort, insertion sort, merge sort, quick sort, selection sort

## Bubble sort - Idea

- starting from the first element, swap adjacent items if they are not in ascending order
- when last item is reached, the last item is the largest
- repeat the above steps for the remaining items to find the second largest item, and so on

## Bubble sort - Example

underlined: being considered  *italic*: sorted

| round | 34 | 10 | 64 | 51 | 32 | 21 |
|---|---|---|---|---|---|---|

10 21 32 34 51 64

## Bubble sort - Example

underlined: being considered  *italic*: sorted

| round | 34 | 10 | 64 | 51 | 32 | 21 | |
|---|---|---|---|---|---|---|---|
| 1 | 10 | 34 | 64 | 51 | 32 | 21 | ← don't need to swap |

## Bubble sort - Example

underlined: being considered  *italic*: sorted

| round | 34 | 10 | 64 | 51 | 32 | 21 | |
|---|---|---|---|---|---|---|---|
| 1 | 10 | 34 | 64 | 51 | 32 | 21 | ← don't need to swap |
| | 10 | 34 | 64 | 51 | 32 | 21 | |

## Bubble sort - Example

underlined: being considered  *italic*: sorted

| round | 34 | 10 | 64 | 51 | 32 | 21 | |
|---|---|---|---|---|---|---|---|
| 1 | 10 | **34** | **64** | 51 | 32 | 21 | ← don't need to swap |
| | 10 | 34 | **64** | **51** | 32 | 21 | |
| | 10 | 34 | 51 | **64** | **32** | 21 | |

## Bubble sort - Example

underlined: being considered  *italic*: sorted

| round | 34 | 10 | 64 | 51 | 32 | 21 | |
|---|---|---|---|---|---|---|---|
| 1 | 10 | **34** | **64** | 51 | 32 | 21 | ← don't need to swap |
| | 10 | 34 | **64** | **51** | 32 | 21 | |
| | 10 | 34 | 51 | **64** | **32** | 21 | |
| | 10 | 34 | 51 | 32 | **64** | **21** | |

## Bubble sort - Example

underlined: being considered  *italic*: sorted

| round | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 10 | **34** | **64** | 51 | 32 | 21 | ← don't need to swap |
| | 10 | 34 | **64** | **51** | 32 | 21 | |
| | 10 | 34 | 51 | **64** | **32** | 21 | |
| | 10 | 34 | 51 | 32 | **64** | **21** | |
| | **10** | **34** | 51 | 32 | 21 | ***64*** | ← don't need to swap |

## Bubble sort - Example

underlined: being considered  *italic*: sorted

| round | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 10 | **34** | **64** | 51 | 32 | 21 | ← don't need to swap |
| | 10 | 34 | **64** | **51** | 32 | 21 | |
| | 10 | 34 | 51 | **64** | **32** | 21 | |
| | 10 | 34 | 51 | 32 | **64** | **21** | |
| | **10** | **34** | 51 | 32 | 21 | ***64*** | ← don't need to swap |
| 2 | 10 | **34** | **51** | 32 | 21 | ***64*** | ← don't need to swap |

## Bubble sort - Example

**underlined**: being considered  ***italic***: sorted

| round | | | | | | | |
|---|---|---|---|---|---|---|---|
| | **34** | **10** | 64 | 51 | 32 | 21 | |
| 1 | 10 | **34** | **64** | 51 | 32 | 21 | ← don't need to swap |
| | 10 | 34 | **64** | **51** | 32 | 21 | |
| | 10 | 34 | 51 | **64** | **32** | 21 | |
| | 10 | 34 | 51 | 32 | **64** | **21** | |
| | **10** | **34** | 51 | 32 | 21 | ***64*** | ← don't need to swap |
| 2 | 10 | **34** | **51** | 32 | 21 | ***64*** | ← don't need to swap |
| | 10 | 34 | **51** | **32** | 21 | ***64*** | |

## Bubble sort - Example

underlined: being considered  *italic*: sorted

| round | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 10 | **34** | **64** | 51 | 32 | 21 | ← don't need to swap |
| | 10 | 34 | **64** | **51** | 32 | 21 | |
| | 10 | 34 | 51 | **64** | **32** | 21 | |
| | 10 | 34 | 51 | 32 | **64** | **21** | |
| | **10** | **34** | 51 | 32 | 21 | ***64*** | ← don't need to swap |
| 2 | 10 | **34** | **51** | 32 | 21 | ***64*** | ← don't need to swap |
| | 10 | 34 | **51** | **32** | 21 | ***64*** | |
| | 10 | 34 | 32 | **51** | **21** | ***64*** | |

## Bubble sort - Example

underlined: being considered  *italic*: sorted

| round | | | | | | | |
|---|---|---|---|---|---|---|---|
| | **34** | **10** | 64 | 51 | 32 | 21 | |
| 1 | 10 | **34** | **64** | 51 | 32 | 21 | ← don't need to swap |
| | 10 | 34 | **64** | **51** | 32 | 21 | |
| | 10 | 34 | 51 | **64** | **32** | 21 | |
| | 10 | 34 | 51 | 32 | **64** | **21** | |
| | **10** | **34** | 51 | 32 | 21 | ***64*** | ← don't need to swap |
| 2 | 10 | **34** | **51** | 32 | 21 | ***64*** | ← don't need to swap |
| | 10 | 34 | **51** | **32** | 21 | ***64*** | |
| | 10 | 34 | 32 | **51** | **21** | ***64*** | |
| | **10** | **34** | 32 | 21 | ***51*** | ***64*** | ← don't need to swap |

## Bubble sort - Example

underlined: being considered  *italic*: sorted

| round | | | | | | | |
|---|---|---|---|---|---|---|---|
| | **34** | **10** | 64 | 51 | 32 | 21 | |
| 1 | 10 | **34** | **64** | 51 | 32 | 21 | ← don't need to swap |
| | 10 | 34 | **64** | **51** | 32 | 21 | |
| | 10 | 34 | 51 | **64** | **32** | 21 | |
| | 10 | 34 | 51 | 32 | **64** | **21** | |
| | **10** | **34** | 51 | 32 | 21 | ***64*** | ← don't need to swap |
| 2 | 10 | **34** | **51** | 32 | 21 | ***64*** | ← don't need to swap |
| | 10 | 34 | **51** | **32** | 21 | ***64*** | |
| | 10 | 34 | 32 | **51** | **21** | ***64*** | |
| | **10** | **34** | 32 | 21 | ***51*** | ***64*** | ← don't need to swap |
| 3 | 10 | **34** | **32** | 21 | ***51*** | ***64*** | |

## Bubble sort - Example

**underlined**: being considered  ***italic***: sorted

| round | 34 | 10 | 64 | 51 | 32 | 21 | |
|---|---|---|---|---|---|---|---|
| 1 | 10 | **34** | ***64*** | 51 | 32 | 21 | ← don't need to swap |
| | 10 | 34 | ***64*** | ***51*** | 32 | 21 | |
| | 10 | 34 | 51 | ***64*** | ***32*** | 21 | |
| | 10 | 34 | 51 | 32 | ***64*** | ***21*** | |
| | ***10*** | ***34*** | 51 | 32 | 21 | ***64*** | ← don't need to swap |
| 2 | 10 | **34** | ***51*** | 32 | 21 | ***64*** | ← don't need to swap |
| | 10 | 34 | ***51*** | ***32*** | 21 | ***64*** | |
| | 10 | 34 | 32 | ***51*** | ***21*** | ***64*** | |
| | ***10*** | ***34*** | 32 | 21 | ***51*** | ***64*** | ← don't need to swap |
| 3 | 10 | **34** | ***32*** | 21 | ***51*** | ***64*** | |
| | 10 | 32 | ***34*** | ***21*** | ***51*** | ***64*** | |

## Bubble sort - Example

**underlined**: being considered  ***italic***: sorted

| round | | | | | | | |
|---|---|---|---|---|---|---|---|
| | **34** | **10** | 64 | 51 | 32 | 21 | |
| 1 | 10 | **34** | **64** | 51 | 32 | 21 | ← don't need to swap |
| | 10 | 34 | **64** | **51** | 32 | 21 | |
| | 10 | 34 | 51 | **64** | **32** | 21 | |
| | 10 | 34 | 51 | 32 | **64** | **21** | |
| | **10** | **34** | 51 | 32 | 21 | ***64*** | ← don't need to swap |
| 2 | 10 | **34** | **51** | 32 | 21 | ***64*** | ← don't need to swap |
| | 10 | 34 | **51** | **32** | 21 | ***64*** | |
| | 10 | 34 | 32 | **51** | **21** | ***64*** | |
| | **10** | **34** | 32 | 21 | ***51*** | ***64*** | ← don't need to swap |
| 3 | 10 | **34** | **32** | 21 | ***51*** | ***64*** | |
| | 10 | 32 | **34** | **21** | ***51*** | ***64*** | |
| | **10** | **32** | 21 | ***34*** | ***51*** | ***64*** | ← don't need to swap |

## Bubble sort - Example

underlined: being considered  *italic*: sorted

| round | | | | | | | |
|---|---|---|---|---|---|---|---|
| | **34** | **10** | 64 | 51 | 32 | 21 | |
| 1 | 10 | **34** | **64** | 51 | 32 | 21 | ← don't need to swap |
| | 10 | 34 | **64** | **51** | 32 | 21 | |
| | 10 | 34 | 51 | **64** | **32** | 21 | |
| | 10 | 34 | 51 | 32 | **64** | **21** | |
| | **10** | **34** | 51 | 32 | 21 | ***64*** | ← don't need to swap |
| 2 | 10 | **34** | **51** | 32 | 21 | ***64*** | ← don't need to swap |
| | 10 | 34 | **51** | **32** | 21 | ***64*** | |
| | 10 | 34 | 32 | **51** | **21** | ***64*** | |
| | **10** | **34** | 32 | 21 | ***51*** | ***64*** | ← don't need to swap |
| 3 | 10 | **34** | **32** | 21 | ***51*** | ***64*** | |
| | 10 | 32 | **34** | **21** | ***51*** | ***64*** | |
| | **10** | **32** | 21 | ***34*** | ***51*** | ***64*** | ← don't need to swap |
| 4 | 10 | **32** | **21** | ***34*** | ***51*** | ***64*** | |

## Bubble sort - Example

**underlined**: being considered  ***italic***: sorted

| round | | | | | | | |
|---|---|---|---|---|---|---|---|
| | **34** | **10** | 64 | 51 | 32 | 21 | |
| 1 | 10 | **34** | ***64*** | 51 | 32 | 21 | ← don't need to swap |
| | 10 | 34 | ***64*** | ***51*** | 32 | 21 | |
| | 10 | 34 | 51 | ***64*** | ***32*** | 21 | |
| | 10 | 34 | 51 | 32 | ***64*** | ***21*** | |
| | ***10*** | ***34*** | 51 | 32 | 21 | ***64*** | ← don't need to swap |
| 2 | 10 | **34** | ***51*** | 32 | 21 | ***64*** | ← don't need to swap |
| | 10 | 34 | ***51*** | ***32*** | 21 | ***64*** | |
| | 10 | 34 | 32 | ***51*** | ***21*** | ***64*** | |
| | ***10*** | ***34*** | 32 | 21 | ***51*** | ***64*** | ← don't need to swap |
| 3 | 10 | **34** | ***32*** | 21 | ***51*** | ***64*** | |
| | 10 | 32 | ***34*** | ***21*** | ***51*** | ***64*** | |
| | ***10*** | ***32*** | 21 | ***34*** | ***51*** | ***64*** | ← don't need to swap |
| 4 | 10 | **32** | ***21*** | ***34*** | ***51*** | ***64*** | |
| | ***10*** | ***21*** | ***32*** | ***34*** | ***51*** | ***64*** | ← don't need to swap |

## Bubble sort - Example

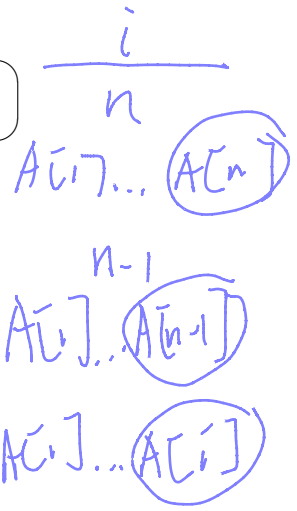
**underlined**: being considered *italic*: sorted

| round | | | | | | | |
|---|---|---|---|---|---|---|---|
| | **34** | **10** | 64 | 51 | 32 | 21 | |
| 1 | 10 | **34** | **64** | 51 | 32 | 21 | ← don't need to swap |
| | 10 | 34 | **64** | **51** | 32 | 21 | |
| | 10 | 34 | 51 | **64** | **32** | 21 | |
| | 10 | 34 | 51 | 32 | **64** | **21** | |
| | **10** | **34** | 51 | 32 | 21 | ***64*** | ← don't need to swap |
| 2 | 10 | **34** | **51** | 32 | 21 | ***64*** | ← don't need to swap |
| | 10 | 34 | **51** | **32** | 21 | ***64*** | |
| | 10 | 34 | 32 | **51** | **21** | ***64*** | |
| | **10** | **34** | 32 | 21 | ***51*** | ***64*** | ← don't need to swap |
| 3 | 10 | **34** | **32** | 21 | ***51*** | ***64*** | |
| | 10 | 32 | **34** | **21** | ***51*** | ***64*** | |
| | **10** | **32** | 21 | ***34*** | ***51*** | ***64*** | ← don't need to swap |
| 4 | 10 | **32** | **21** | ***34*** | ***51*** | ***64*** | |
| | **10** | **21** | ***32*** | ***34*** | ***51*** | ***64*** | ← don't need to swap |
| 5 | 10 | ***21*** | ***32*** | ***34*** | ***51*** | ***64*** | |

## Bubble sort algorithm

for i ← **n** downto **2** do // downto: i automatically dec. by 1 every time

// move the largest number in $(A[1] \cdots A[i])$ to $A[i]$ by
// swapping neighbouring numbers if they are not in correct order



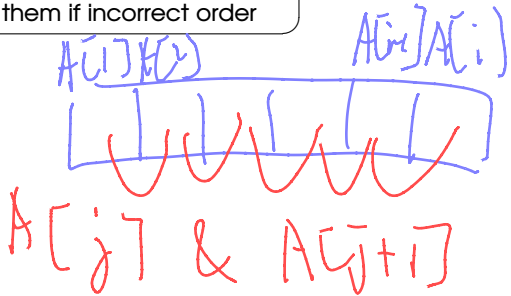
## Bubble sort algorithm

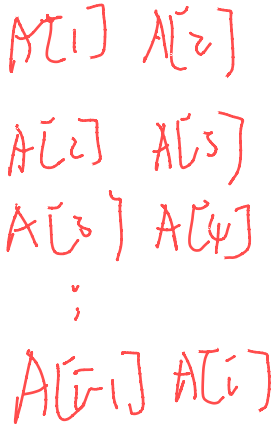
for i ← **n** downto **2** do // downto: i automatically dec. by 1 every time

// move the largest number in $(A[1] \cdots A[i])$ to $A[i]$ by
// swapping neighbouring numbers if they are not in correct order

for j ← **1** to **i-1** do // to: j automatically inc. by 1 every time

// compare $A[j]$ and $A[j+1]$
// swap them if incorrect order





## Bubble sort algorithm

for i $\leftarrow$ **n** downto **2** do  // downto: i automatically dec. by 1 every time

// move the largest number in $(A[1] \cdots A[i])$ to $A[i]$ by
// swapping neighbouring numbers if they are not in correct order

for j $\leftarrow$ **1** to **i-1** do  // to: j automatically inc. by 1 every time

// compare $A[j]$ and $A[j+1]$
// swap them if incorrect order

if $(A[j] > A[j+1])$ then
  swap $A[j]$ & $A[j+1]$

## Bubble sort algorithm

for i ← **n** downto **2** do  // downto: i automatically dec. by 1 every time

// move the largest number in $(A[1] \cdots A[i])$ to $A[i]$ by
// swapping neighbouring numbers if they are not in correct order

for j ← **1** to **i-1** do  // to: j automatically inc. by 1 every time

// compare $A[j]$ and $A[j+1]$
// swap them if incorrect order

if $(A[j] > A[j+1])$ then
  swap $A[j]$ & $A[j+1]$

How to swap two variables?

## Bubble sort algorithm - Swapping two variables

- What's wrong with this?

$$x \leftarrow y$$
$$y \leftarrow x$$

## Bubble sort algorithm - Swapping two variables

- What's wrong with this?

$$x \leftarrow y$$
$$y \leftarrow x$$

Both variables will store the original value of y

## Bubble sort algorithm - Swapping two variables

- What's wrong with this?

$x \leftarrow y$

$y \leftarrow x$

| x | y |
|---|---|
| 4 | 6 |
| 6 | 6 |
| 6 | 6 |

Both variables will store the original value of y

## Bubble sort algorithm - Swapping two variables

- What's wrong with this?

$$x \leftarrow y$$
$$y \leftarrow x$$

| x | y |
|---|---|
| 4 | 6 |
| 6 | 6 |
| 6 | 6 |

  Both variables will store the original value of y

- Using a temporary variable

## Bubble sort algorithm - Swapping two variables

- What's wrong with this?

$$x \leftarrow y$$
$$y \leftarrow x$$

| x | y |
|---|---|
| 4 | 6 |
| 6 | 6 |
| 6 | 6 |

Both variables will store the original value of y

- Using a temporary variable

$$tmp \leftarrow x$$

## Bubble sort algorithm - Swapping two variables

- What's wrong with this?

$$x \leftarrow y$$
$$y \leftarrow x$$

| x | y |
|---|---|
| 4 | 6 |
| 6 | 6 |
| 6 | 6 |

Both variables will store the original value of y

- Using a temporary variable

$$tmp \leftarrow x$$
$$x \leftarrow y$$

## Bubble sort algorithm - Swapping two variables

- What's wrong with this?

$$x \leftarrow y$$
$$y \leftarrow x$$

| x | y |
|---|---|
| 4 | 6 |
| 6 | 6 |
| 6 | 6 |

Both variables will store the original value of y

- Using a temporary variable

$$tmp \leftarrow x$$
$$x \leftarrow y$$
$$y \leftarrow tmp$$

## Bubble sort algorithm - Swapping two variables

- What's wrong with this?

$x \leftarrow y$
$y \leftarrow x$

| x | y |
|---|---|
| 4 | 6 |
| 6 | 6 |
| 6 | 6 |

Both variables will store the original value of y

- Using a temporary variable

$tmp \leftarrow x$
$x \leftarrow y$
$y \leftarrow tmp$

| x | y | tmp |
|---|---|---|
| 4 | 6 | |
| 4 | 6 | 4 |
| 6 | 6 | 4 |
| 6 | 4 | 4 |

## Bubble sort algorithm - Swapping two variables

- What's wrong with this?

$x \leftarrow y$
$y \leftarrow x$

| x | y |
|---|---|
| 4 | 6 |
| 6 | 6 |
| 6 | 6 |

Both variables will store the original value of y

- Using a temporary variable

$tmp \leftarrow x$
$x \leftarrow y$
$y \leftarrow tmp$

| x | y | tmp |
|---|---|---|
| 4 | 6 | |
| 4 | 6 | 4 |
| 6 | 6 | 4 |
| 6 | 4 | 4 |

- Mathematically, is it possible to swap without using extra storage?

## Bubble sort algorithm - Swapping two variables

- What's wrong with this?

$x \leftarrow y$
$y \leftarrow x$

| x | y |
|---|---|
| 4 | 6 |
| 6 | 6 |
| 6 | 6 |

Both variables will store the original value of y

- Using a temporary variable

$tmp \leftarrow x$
$x \leftarrow y$
$y \leftarrow tmp$

| x | y | tmp |
|---|---|---|
| 4 | 6 | |
| 4 | 6 | 4 |
| 6 | 6 | 4 |
| 6 | 4 | 4 |

- Mathematically, is it possible to swap without using extra storage?

$x \leftarrow x + y$

## Bubble sort algorithm - Swapping two variables

- What's wrong with this?

$x \leftarrow y$
$y \leftarrow x$

| x | y |
|---|---|
| 4 | 6 |
| 6 | 6 |
| 6 | 6 |

Both variables will store the original value of y

- Using a temporary variable

$tmp \leftarrow x$
$x \leftarrow y$
$y \leftarrow tmp$

| x | y | tmp |
|---|---|---|
| 4 | 6 | |
| 4 | 6 | 4 |
| 6 | 6 | 4 |
| 6 | 4 | 4 |

- Mathematically, is it possible to swap without using extra storage?

$x \leftarrow x + y$
$y \leftarrow x - y$

## Bubble sort algorithm - Swapping two variables

- What's wrong with this?

$x \leftarrow y$
$y \leftarrow x$

| x | y |
|---|---|
| 4 | 6 |
| 6 | 6 |
| 6 | 6 |

Both variables will store the original value of y

- Using a temporary variable

$tmp \leftarrow x$
$x \leftarrow y$
$y \leftarrow tmp$

| x | y | tmp |
|---|---|---|
| 4 | 6 | |
| 4 | 6 | 4 |
| 6 | 6 | 4 |
| 6 | 4 | 4 |

- Mathematically, is it possible to swap without using extra storage?

$x \leftarrow x + y$
$y \leftarrow x - y$
$x \leftarrow x - y$

## Bubble sort algorithm - Swapping two variables

- What's wrong with this?

$x \leftarrow y$
$y \leftarrow x$

| x | y |
|---|---|
| 4 | 6 |
| 6 | 6 |
| 6 | 6 |

```
x <- x XOR y
y <- x XOR y
x <- x XOR y
```

Both variables will store the original value of y

- Using a temporary variable

$tmp \leftarrow x$
$x \leftarrow y$
$y \leftarrow tmp$

| x | y | tmp |
|---|---|---|
| 4 | 6 | |
| 4 | 6 | 4 |
| 6 | 6 | 4 |
| 6 | 4 | 4 |

```
x 110
y 100
x 010
y 110
x 100
```

- Mathematically, is it possible to swap without using extra storage?

$x \leftarrow x + y$
$y \leftarrow x - y$
$x \leftarrow x - y$

| x | y |
|---|---|
| 4 | 6 |
| 10 | 6 |
| 10 | 4 |
| 6 | 4 |

Why aren't we doing this in programs? `potential overflow`

## Bubble sort algorithm

for $i \leftarrow n$ downto 2 do
  for $j \leftarrow 1$ to $i - 1$ do
    if $A[j] > A[j+1]$ then
      swap $A[j]$ & $A[j+1]$

## Bubble sort algorithm

for $i \leftarrow n$ downto 2 do
  for $j \leftarrow 1$ to $i - 1$ do
    if $A[j] > A[j+1]$ then
      swap $A[j]$ & $A[j+1]$

| | 34 | 10 | 64 | 51 | 32 | 21 |
|---|---|---|---|---|---|---|
| i=6 | 34 | 10 | ← j=1 | | | |

## Bubble sort algorithm

for $i \leftarrow n$ downto 2 do
  for $j \leftarrow 1$ to $i - 1$ do
    if $A[j] > A[j+1]$ then
      swap $A[j]$ & $A[j+1]$

| | | | | | | |
|---|---|---|---|---|---|---|
| | 34 | 10 | 64 | 51 | 32 | 21 |
| i=6 | 34 | 10 | ← j=1 | | | |
| | 10 | 34 | 64 | ← j=2 | | |

## Bubble sort algorithm

for $i \leftarrow n$ downto 2 do
  for $j \leftarrow 1$ to $i - 1$ do
    if $A[j] > A[j+1]$ then
      swap $A[j]$ & $A[j+1]$

| | | | | | | |
|---|---|---|---|---|---|---|
| | 34 | 10 | 64 | 51 | 32 | 21 |
| i=6 | 34 | 10 | ← j=1 | | | |
| | 10 | 34 | 64 | ← j=2 | | |
| | 10 | 34 | 64 | 51 | ← j=3 | |

## Bubble sort algorithm

for $i \leftarrow n$ downto 2 do
  for $j \leftarrow 1$ to $i - 1$ do
    if $A[j] > A[j+1]$ then
      swap $A[j]$ & $A[j+1]$

| | | | | | | |
|---|---|---|---|---|---|---|
| | 34 | 10 | 64 | 51 | 32 | 21 |
| i=6 | 34 | 10 | ← j=1 | | | |
| | 10 | 34 | 64 | ← j=2 | | |
| | 10 | 34 | 64 | 51 | ← j=3 | |
| | 10 | 34 | 51 | 64 | 32 | ← j=4 |

## Bubble sort algorithm

for $i \leftarrow n$ downto 2 do
  for $j \leftarrow 1$ to $i - 1$ do
    if $A[j] > A[j+1]$ then
      swap $A[j]$ & $A[j+1]$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | 34 | 10 | 64 | 51 | 32 | 21 | |
| i=6 | 34 | 10 | ← j=1 | | | | |
| | 10 | 34 | 64 | ← j=2 | | | |
| | 10 | 34 | 64 | 51 | ← j=3 | | |
| | 10 | 34 | 51 | 64 | 32 | ← j=4 | |
| | 10 | 34 | 51 | 32 | 64 | 21 | ← j=5 |

## Bubble sort algorithm

for $i \leftarrow n$ downto 2 do
  for $j \leftarrow 1$ to $i - 1$ do
    if $A[j] > A[j+1]$ then
      swap $A[j]$ & $A[j+1]$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | 34 | 10 | 64 | 51 | 32 | 21 | |
| i=6 | 34 | 10 | ← j=1 | | | | |
| | 10 | 34 | 64 | ← j=2 | | | |
| | 10 | 34 | 64 | 51 | ← j=3 | | |
| | 10 | 34 | 51 | 64 | 32 | ← j=4 | |
| | 10 | 34 | 51 | 32 | 64 | 21 | ← j=5 |
| i=5 | 10 | 34 | ← j=1 | | | | |

## Bubble sort algorithm

for $i \leftarrow n$ downto 2 do
  for $j \leftarrow 1$ to $i - 1$ do
    if $A[j] > A[j+1]$ then
      swap $A[j]$ & $A[j+1]$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | 34 | 10 | 64 | 51 | 32 | 21 | |
| i=6 | 34 | 10 | ← j=1 | | | | |
| | 10 | 34 | 64 | ← j=2 | | | |
| | 10 | 34 | 64 | 51 | ← j=3 | | |
| | 10 | 34 | 51 | 64 | 32 | ← j=4 | |
| | 10 | 34 | 51 | 32 | 64 | 21 | ← j=5 |
| i=5 | 10 | 34 | ← j=1 | | | | |
| | 10 | 34 | 51 | ← j=2 | | | |

## Bubble sort algorithm

for $i \leftarrow n$ downto 2 do
  for $j \leftarrow 1$ to $i - 1$ do
    if $A[j] > A[j+1]$ then
      swap $A[j]$ & $A[j+1]$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | 34 | 10 | 64 | 51 | 32 | 21 | |
| i=6 | 34 | 10 | ← j=1 | | | | |
| | 10 | 34 | 64 | ← j=2 | | | |
| | 10 | 34 | 64 | 51 | ← j=3 | | |
| | 10 | 34 | 51 | 64 | 32 | ← j=4 | |
| | 10 | 34 | 51 | 32 | 64 | 21 | ← j=5 |
| i=5 | 10 | 34 | ← j=1 | | | | |
| | 10 | 34 | 51 | ← j=2 | | | |
| | 10 | 34 | 51 | 32 | ← j=3 | | |

## Bubble sort algorithm

for $i \leftarrow n$ downto 2 do
  for $j \leftarrow 1$ to $i - 1$ do
    if $A[j] > A[j+1]$ then
      swap $A[j]$ & $A[j+1]$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | 34 | 10 | 64 | 51 | 32 | 21 | |
| i=6 | 34 | 10 | ← j=1 | | | | |
| | 10 | 34 | 64 | ← j=2 | | | |
| | 10 | 34 | 64 | 51 | ← j=3 | | |
| | 10 | 34 | 51 | 64 | 32 | ← j=4 | |
| | 10 | 34 | 51 | 32 | 64 | 21 | ← j=5 |
| i=5 | 10 | 34 | ← j=1 | | | | |
| | 10 | 34 | 51 | ← j=2 | | | |
| | 10 | 34 | 51 | 32 | ← j=3 | | |
| | 10 | 34 | 32 | 51 | 21 | ← j=4 | |

## Bubble sort algorithm - Using nested while-loops

```
for i ← n downto 2 do
    for j ← 1 to i − 1 do
        if A[j] > A[j + 1] then
            swap A[j] & A[j + 1]
```

```
i ← n
while n ≥ 2 do
begin
    j ← 1
    while j ≤ i − 1 do
    begin
        if A[j] > A[j + 1] then
            swap A[j] & A[j + 1]
        j ← j + 1
    end
    i ← i − 1
end
```

## Bubble sort algorithm - Time complexity

The algorithm consists of a nested for-loop. For each iteration of the outer i-loop, there is an inner j-loop.

```
for i ← n downto 2 do
    for j ← 1 to i − 1 do
        if A[j] > A[j + 1] then
            swap A[j] & A[j + 1]
```

## Bubble sort algorithm - Time complexity

The algorithm consists of a nested for-loop. For each iteration of the outer i-loop, there is an inner j-loop.

$O((n-1)^2)$

$O(n!)$

$O(n^2)$

```
for i ← n downto 2 do
    for j ← 1 to i − 1 do
        if A[j] > A[j + 1] then
            swap A[j] & A[j + 1]
```

| i | # of > comparisons |
|---|---|
| n | n-1 |
| n-1 | n-2 |
| ⋮ | ⋮ |
| 2 | 1 |

## Bubble sort algorithm - Time complexity

The algorithm consists of a nested for-loop. For each iteration of the outer i-loop, there is an inner j-loop.

```
for i ← n downto 2 do
    for j ← 1 to i − 1 do
        if A[j] > A[j + 1] then
            swap A[j] & A[j + 1]
```

$$
\begin{aligned}
& \text{Total number of comparisons} \\
= \; & (n-1) + (n-2) + \cdots + 2 + 1 \\
= \; & \frac{n(n-1)}{2}
\end{aligned}
$$

| i | # of > comparisons |
|---|---|
| n | n-1 |
| n-1 | n-2 |
| ⋮ | ⋮ |
| 2 | 1 |

## Bubble sort algorithm - Time complexity

The algorithm consists of a nested for-loop. For each iteration of the outer i-loop, there is an inner j-loop.

```
for i ← n downto 2 do
    for j ← 1 to i − 1 do
        if A[j] > A[j + 1] then
            swap A[j] & A[j + 1]
```

Total number of comparisons

$$= (n-1) + (n-2) + \cdots + 2 + 1$$

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

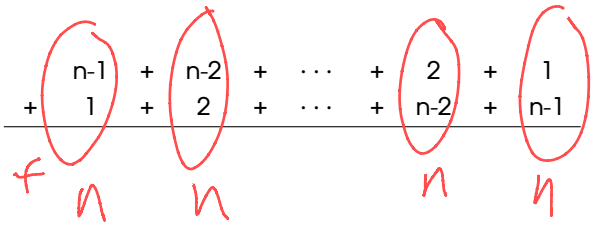
$O(n^2)$-time

| i | # of > comparisons |
|---|---|
| n | n-1 |
| n-1 | n-2 |
| ⋮ | ⋮ |
| 2 | 1 |

## Calculating Sum from 1 to n-1

$$n\text{-}1 \quad + \quad n\text{-}2 \quad + \quad \cdots \quad + \quad 2 \quad + \quad 1$$

## Calculating Sum from 1 to n-1




$$\frac{n \times (n-1)}{2}$$

## Calculating Sum from 1 to n-1

$$
\begin{array}{cccccccccc}
 & n\text{-}1 & + & n\text{-}2 & + & \cdots & + & 2 & + & 1 \\
+ & 1 & + & 2 & + & \cdots & + & n\text{-}2 & + & n\text{-}1 \\
\hline
= & n & + & n & + & \cdots & + & n & + & n
\end{array}
$$

## Calculating Sum from 1 to n-1

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | n-1 | + | n-2 | + | $\cdots$ | + | 2 | + | 1 |
| + | 1 | + | 2 | + | $\cdots$ | + | n-2 | + | n-1 |
| = | n | + | n | + | $\cdots$ | + | n | + | n |

Therefore, $(n-1)+(n-2)+\cdots+2+1=\frac{n(n-1)}{2}$

$\frac{n^2}{2}-\frac{n}{2}$ $O(??)$ $O(n^2)$

## Time Complexity of Bubble Sort Algorithm

Which of the following statements is/are correct? Assume we have $n$ numbers to be sorted by bubble sort algorithm.

In the best case, bubble sort takes 0 swap operation.

In the worst case, bubble sort takes $O(n^2)$ swap operations.

Bubble sort takes $O(n^2)$ comparisons.

The worst case time complexity of bubble sort is $O(n^2)$.

Summary: Bubble Sort Algorithm with Array

Next: Bubble Sort Algorithm with Linked List

## For note taking