Programming Session 2

Jan Tekülve

jan.tekuelve@ini.rub.de

Computer Science and Mathematics
Preparatory Course

30.09.2019

Ask for a correct user input

- Sometimes a specific user input is required

```
userIn = input("Please type exit! ")
while not userIn == "exit" :
        userIn = input("Please type exit! ")
```

Ask for a correct user input

- Sometimes a specific user input is required

```
userIn = input("Please type exit! ")
while not userIn == "exit" :
        userIn = input("Please type exit! ")
```

- The input might allow a range of options

```
userIn = input("Please choose Left or Right: ")
while not (userIn == "Left" or userIn == "Right"):
        userIn = input("Please choose Left or Right: ")
```

Variations of the For-Loop

- The range function has an optional stepsize parameter

```
myList = ["A","B","C","D","E","F"]
#Print every second element of a list
for i in range(0,len(myList),2):
        print(myList[i])
#This prints A C E
```

Variations of the For-Loop

- The range function has an optional stepsize parameter

```
myList = ["A","B","C","D","E","F"]
#Print every second element of a list
for i in range(0,len(myList),2):
        print(myList[i])
#This prints A C E
```

- One can even go through the list in reverse

```
#From len(myList)-1 to 0 with stepsize -1
for i in range(len(myList)-1,-1,-1):
        print(myList[i])
#This prints F E D C B A
```

Dissecting Strings

- Split a sentence into words

```
mySentence = "Hello I am a Sentence"
words = mySentence.split(" ") # words is a list
# ["Hello", "I", "am" , "a", "Sentence"]
```

Dissecting Strings

- Split a sentence into words

```
mySentence = "Hello I am a Sentence"
words = mySentence.split(" ") # words is a list
# ["Hello", "I", "am" , "a", "Sentence"]
```

- Split a word into letters

```
word = "Hello"
#The list typecast converts strings to lists
letters = list(word) #["H","e","l","l","o"]
```

Dissecting Strings

- Split a sentence into words

```
mySentence = "Hello I am a Sentence"
words = mySentence.split(" ") # words is a list
# ["Hello", "I", "am" , "a", "Sentence"]
```

- Split a word into letters

```
word = "Hello"
#The list typecast converts strings to lists
letters = list(word) #["H","e","l","l","o"]
```

- Use the “in” operator to check if an element is in a list

```
if "e" in letters:
        print("The letter 'e' is in the list.")
```

Exchange Variable Values

- How to exchange two variable values?

```
FirstPlace = "Schumacher"
SecondPlace = "Lauda"
```

Exchange Variable Values

- How to exchange two variable values?

```
FirstPlace = "Schumacher"
SecondPlace = "Lauda"
```

- Now Lauda overtakes Schumacher

```
FirstPlace = SecondPlace # FirstPlace = "Lauda"
SecondPlace = Firstplace # SecondPlace = "Lauda" !!!
```

Exchange Variable Values

- How to exchange two variable values?

```
FirstPlace = "Schumacher"
SecondPlace = "Lauda"
```

- Now Lauda overtakes Schumacher

```
FirstPlace = SecondPlace # FirstPlace = "Lauda"
SecondPlace = Firstplace # SecondPlace = "Lauda" !!!
```

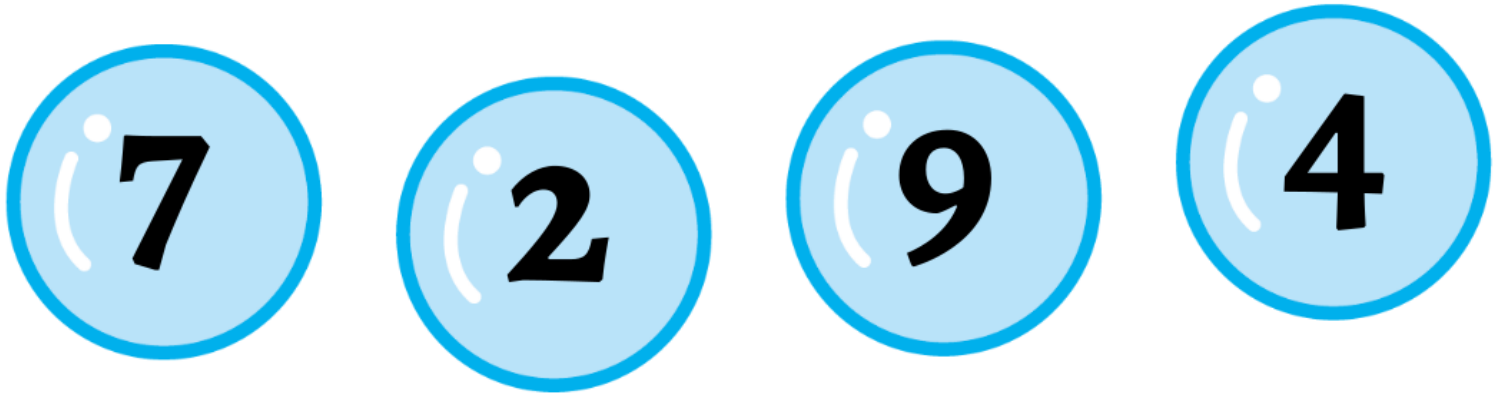
- A helper variable is required

```
helper = FirstPlace # helper = "Schumacher"
FirstPlace = SecondPlace # FirstPlace = "Lauda"
SecondPlace = helper # SecondPlace = "Schumacher"
```

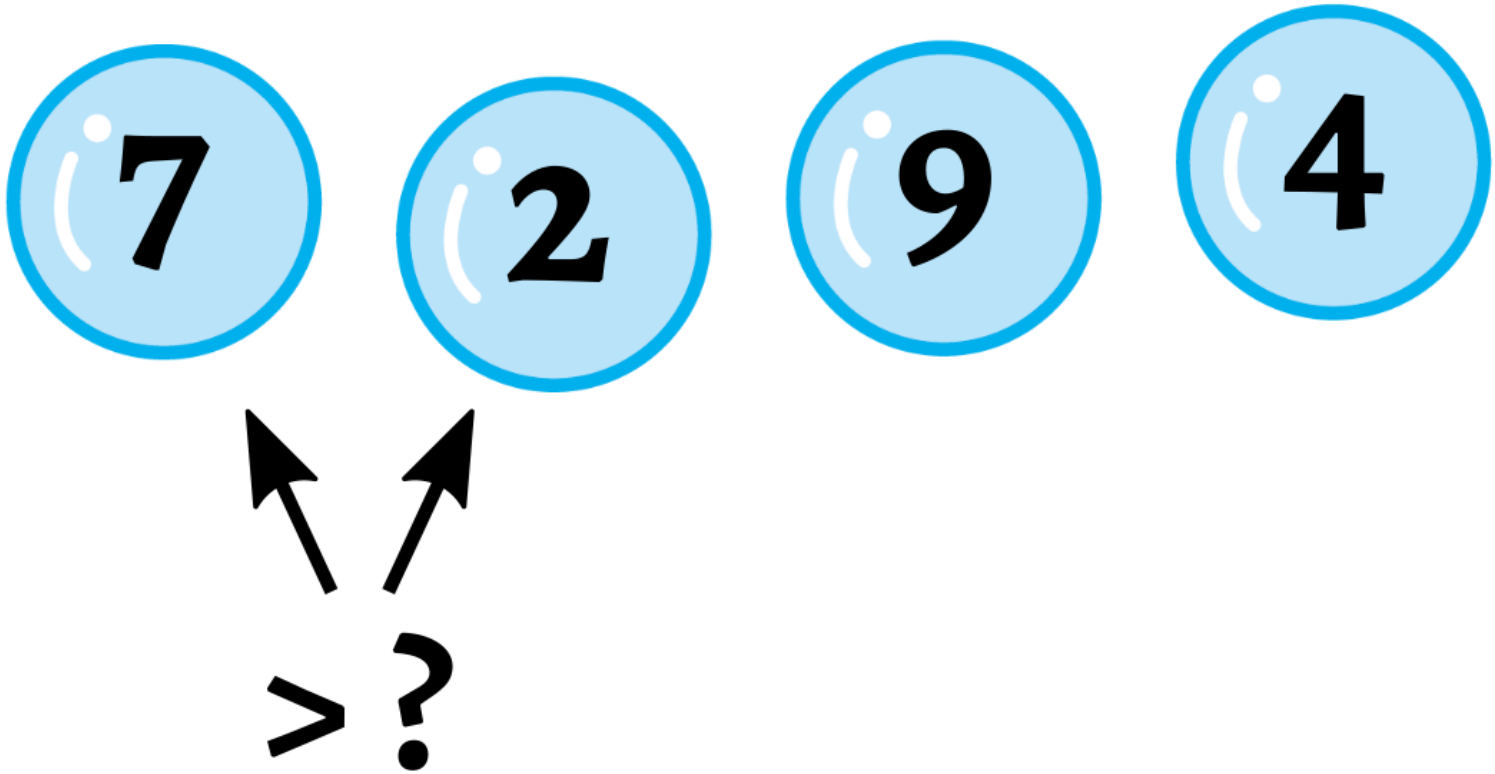
The Bubble Sort Algorithm

$$List = [7,2,9,4]$$

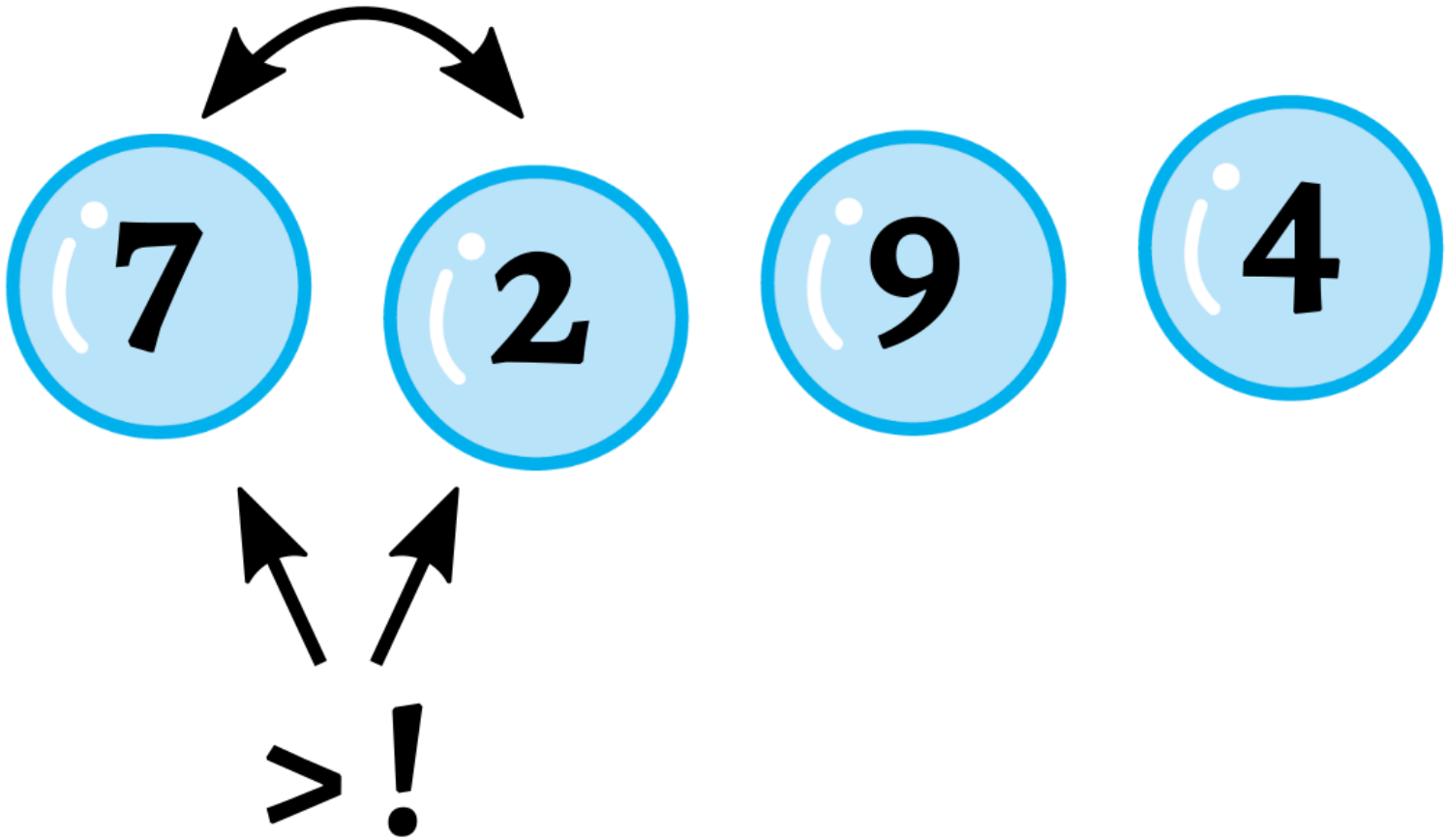
The Bubble Sort Algorithm



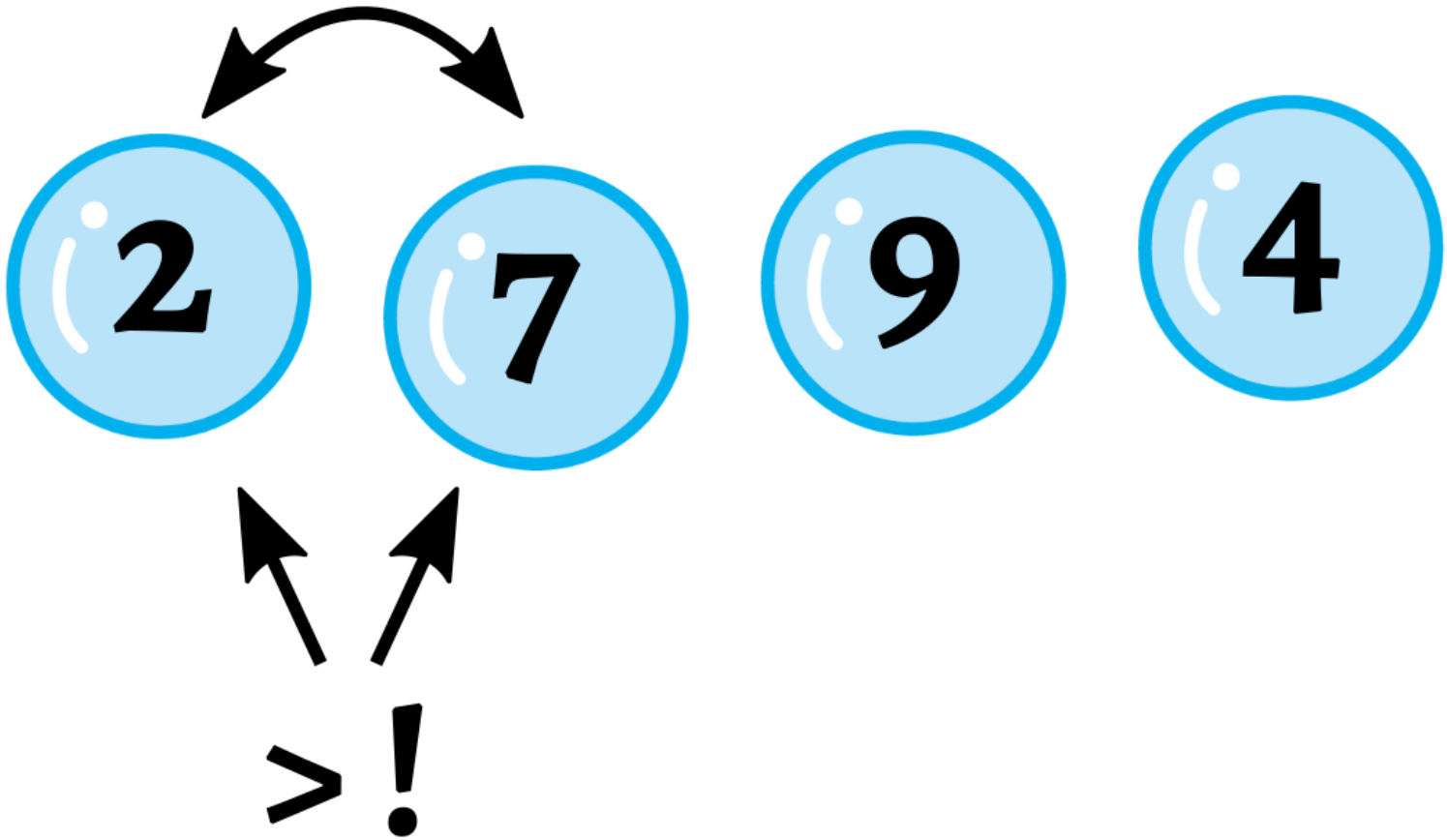
The Bubble Sort Algorithm



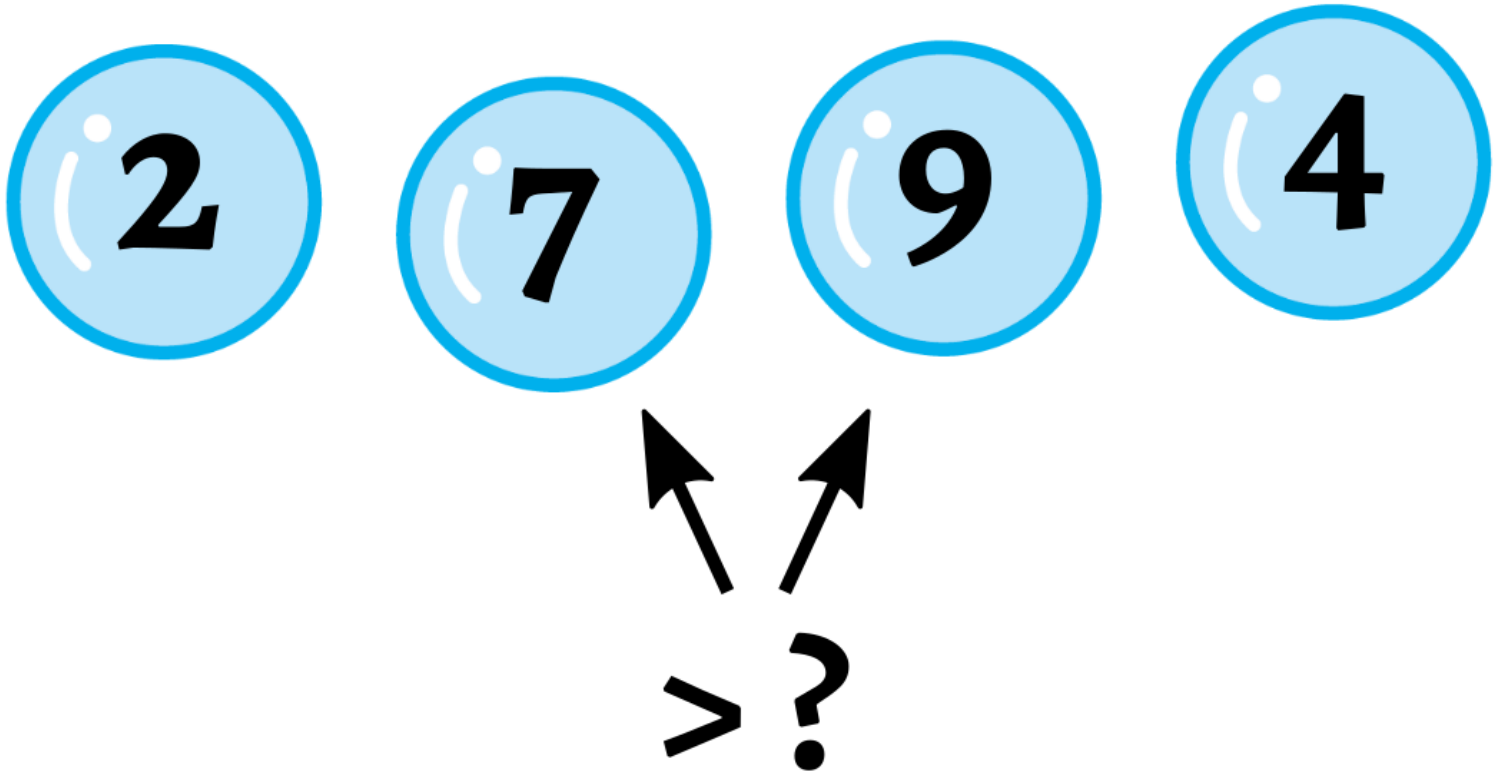
The Bubble Sort Algorithm



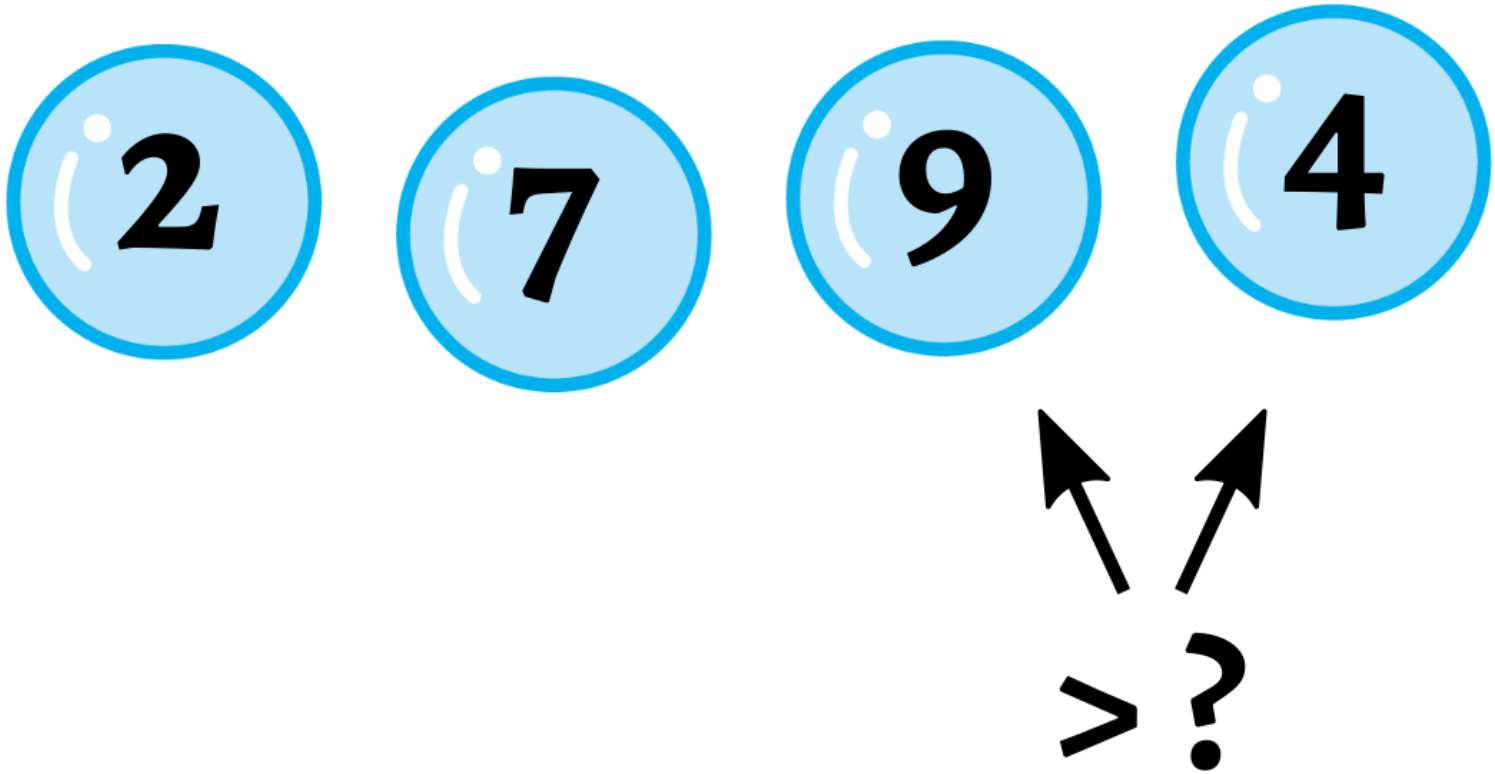
The Bubble Sort Algorithm



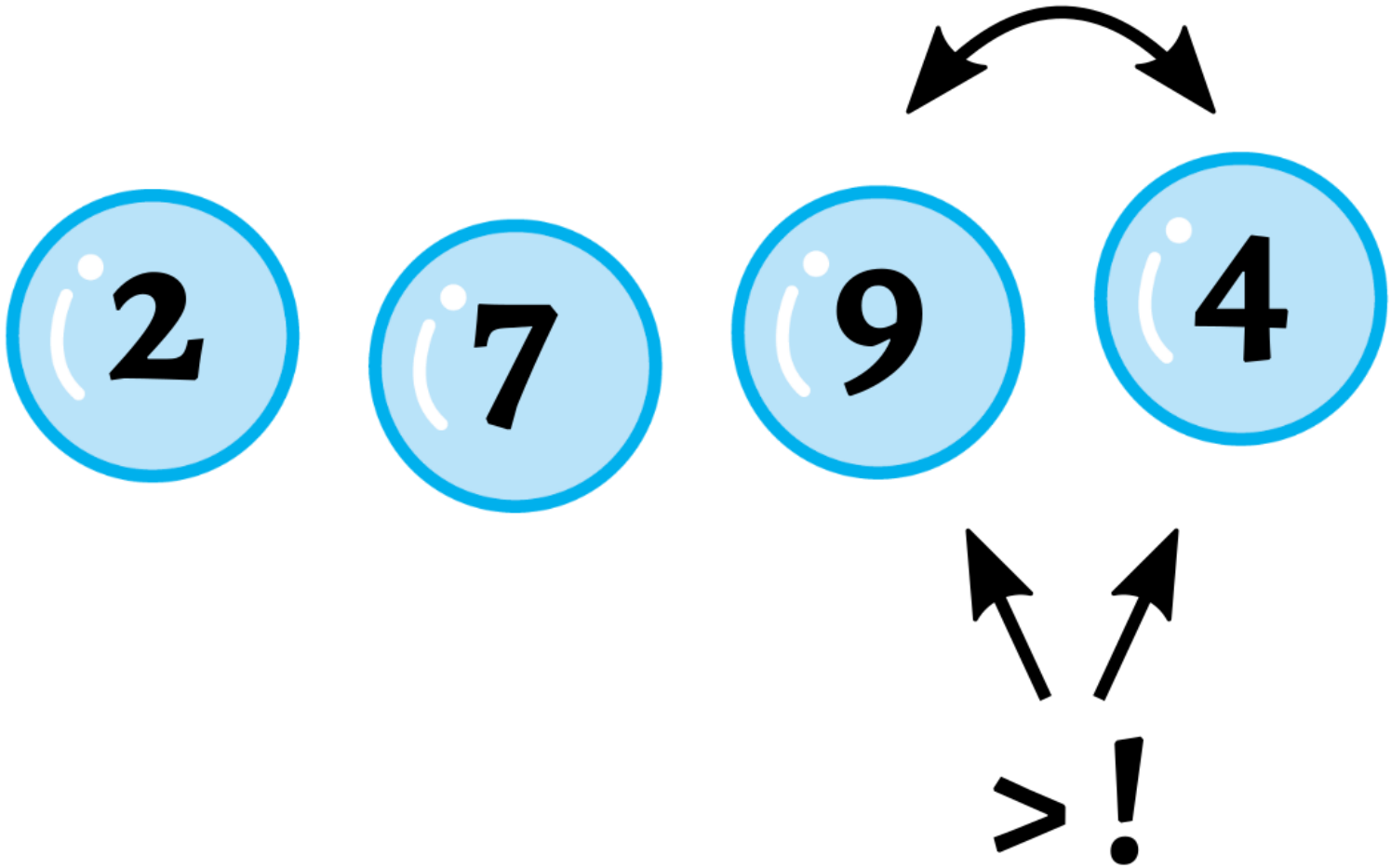
The Bubble Sort Algorithm



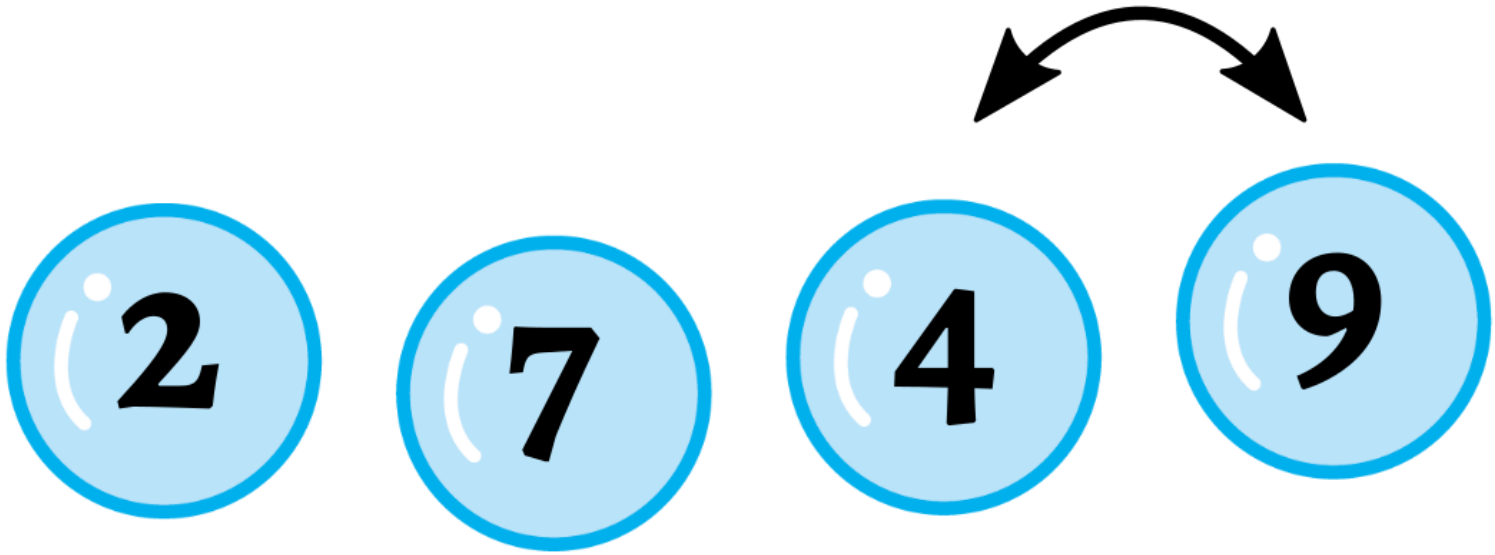
The Bubble Sort Algorithm



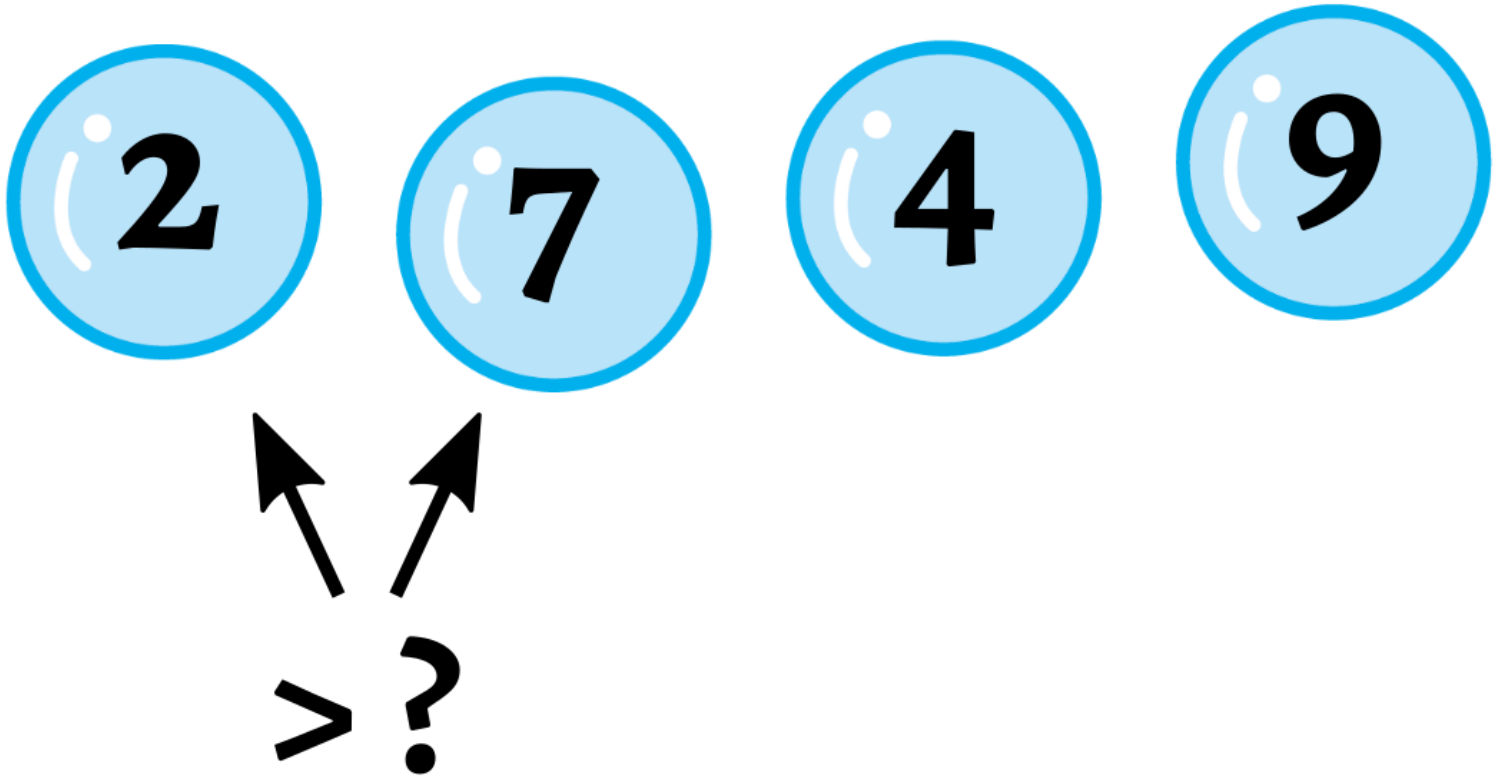
The Bubble Sort Algorithm



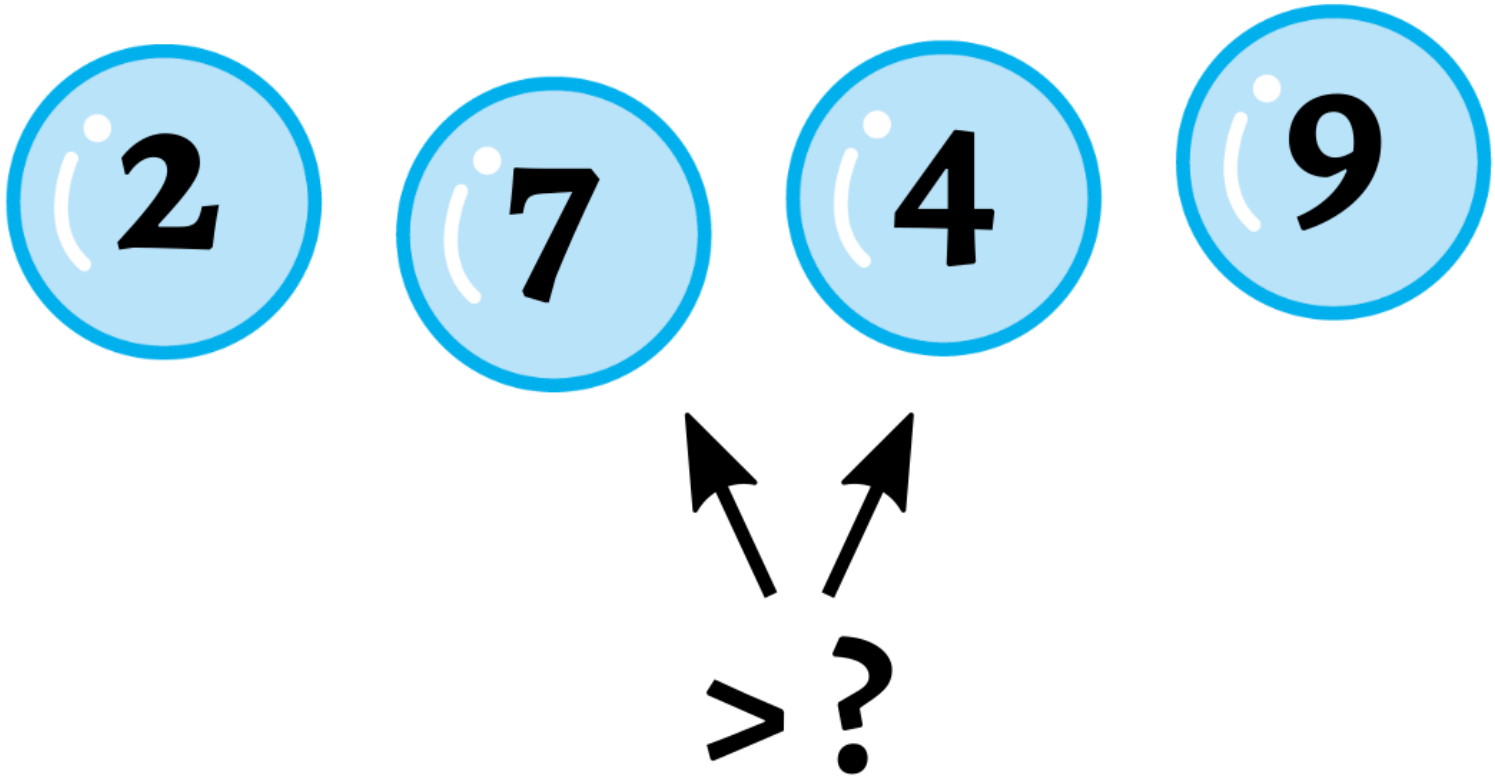
The Bubble Sort Algorithm



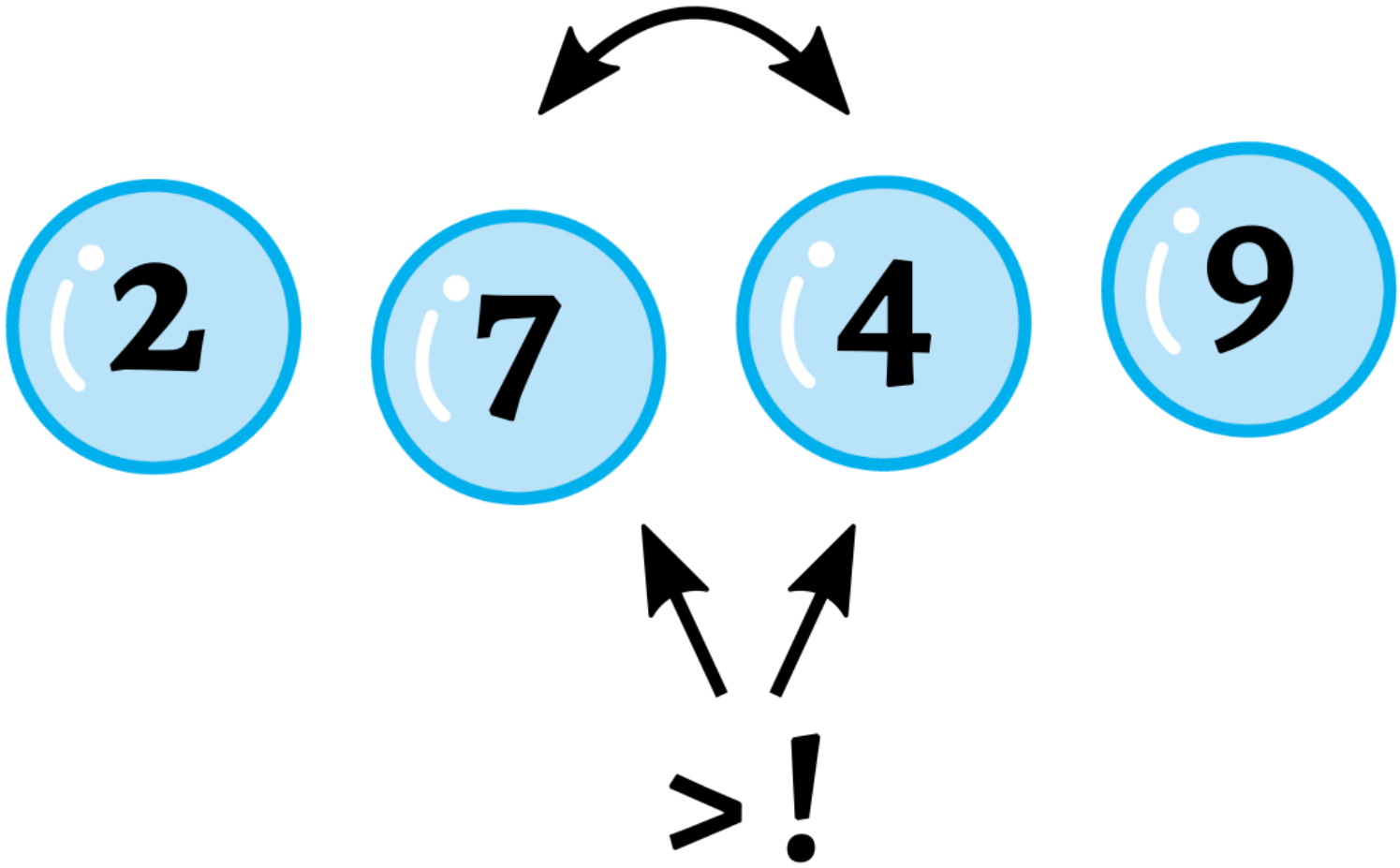
The Bubble Sort Algorithm



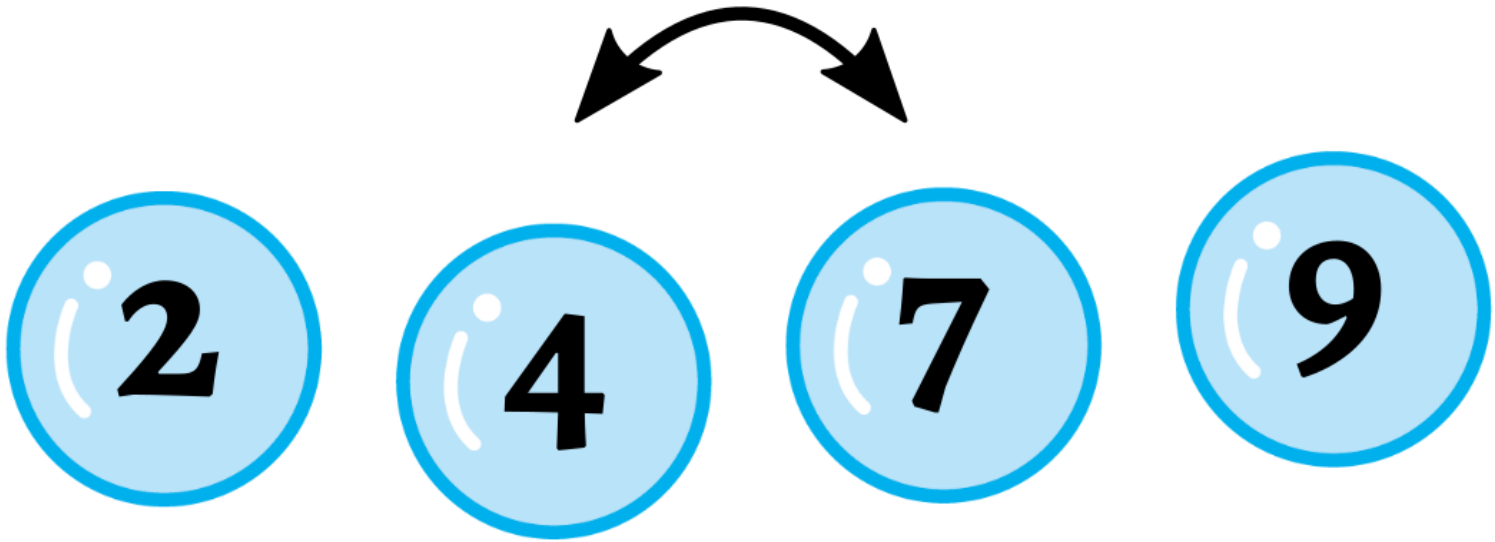
The Bubble Sort Algorithm



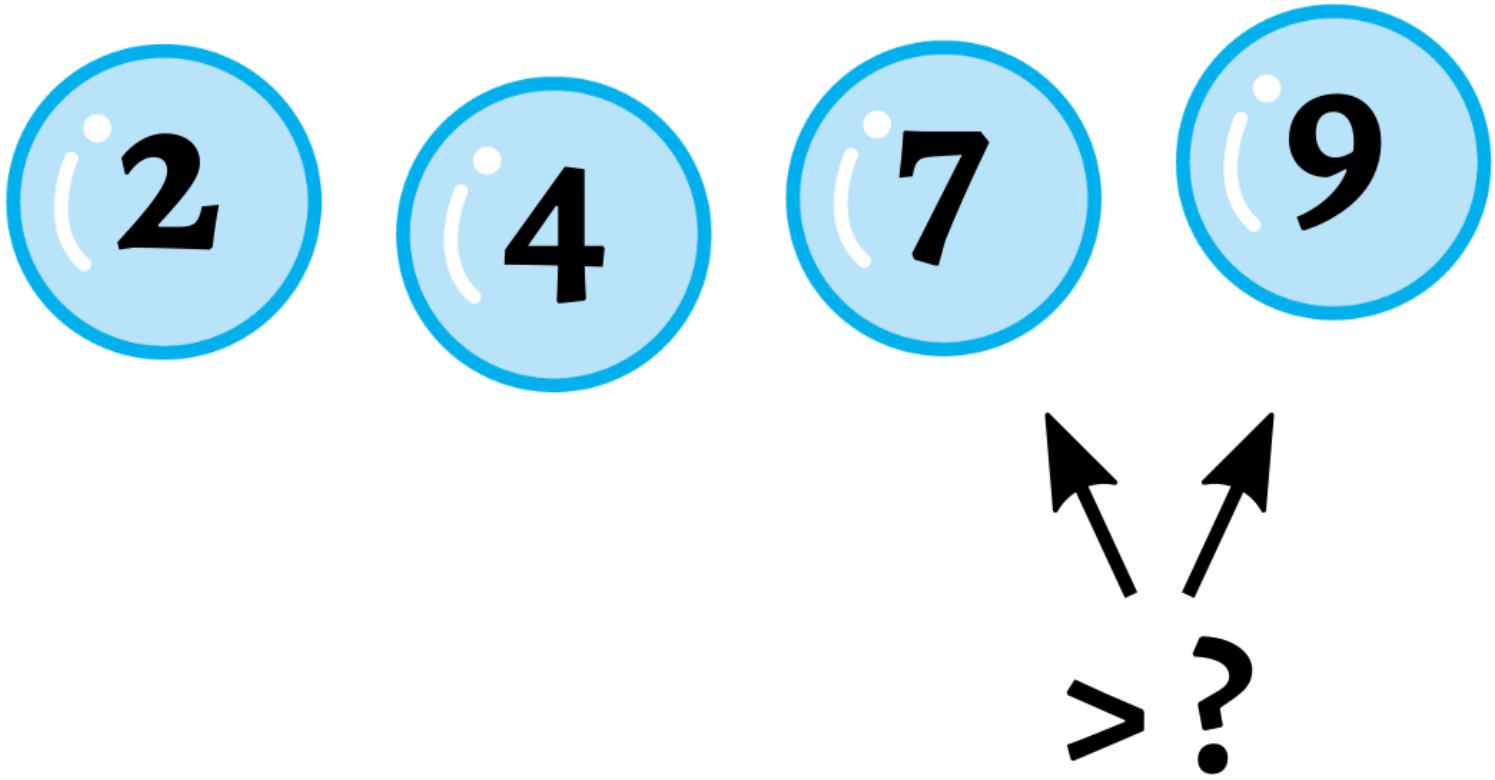
The Bubble Sort Algorithm



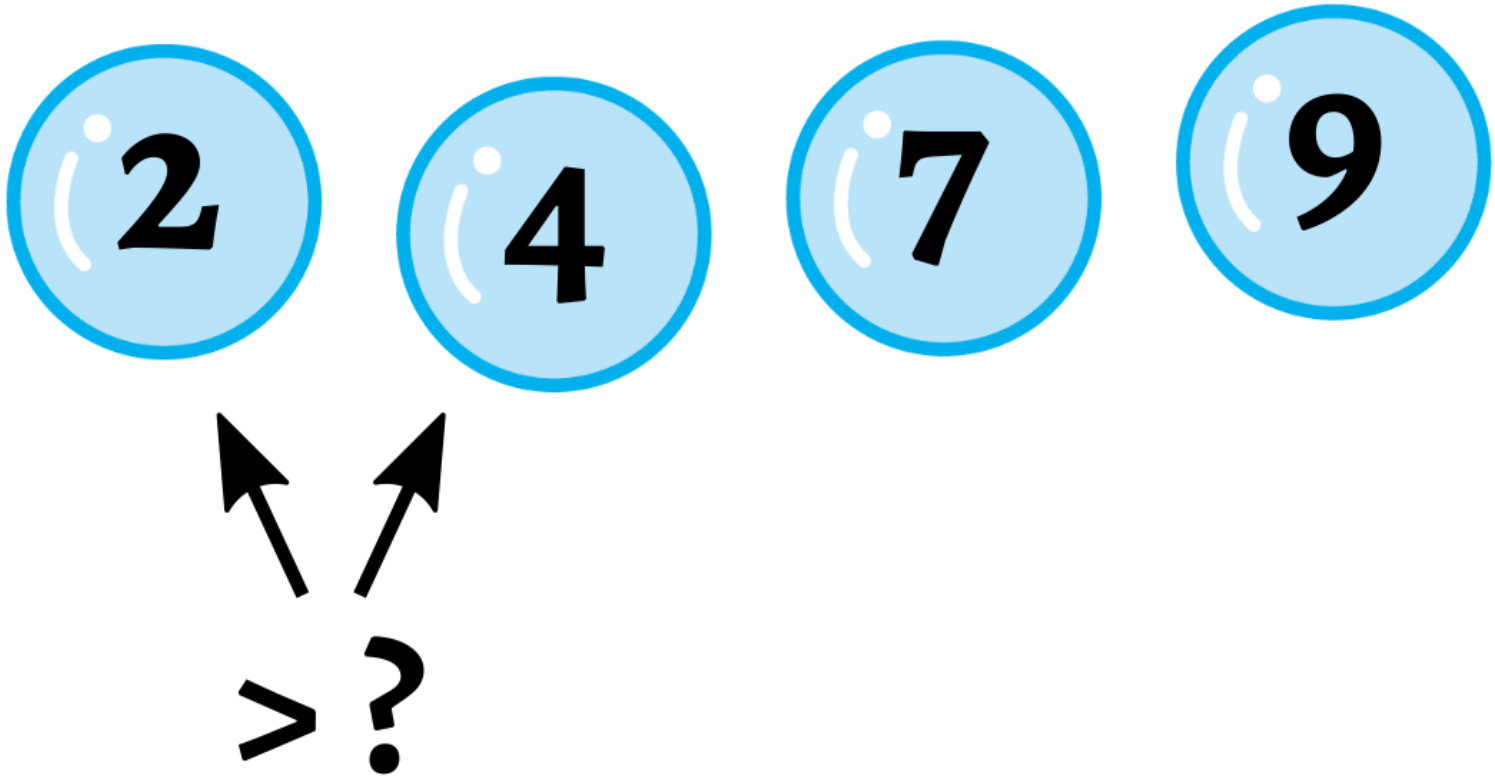
The Bubble Sort Algorithm



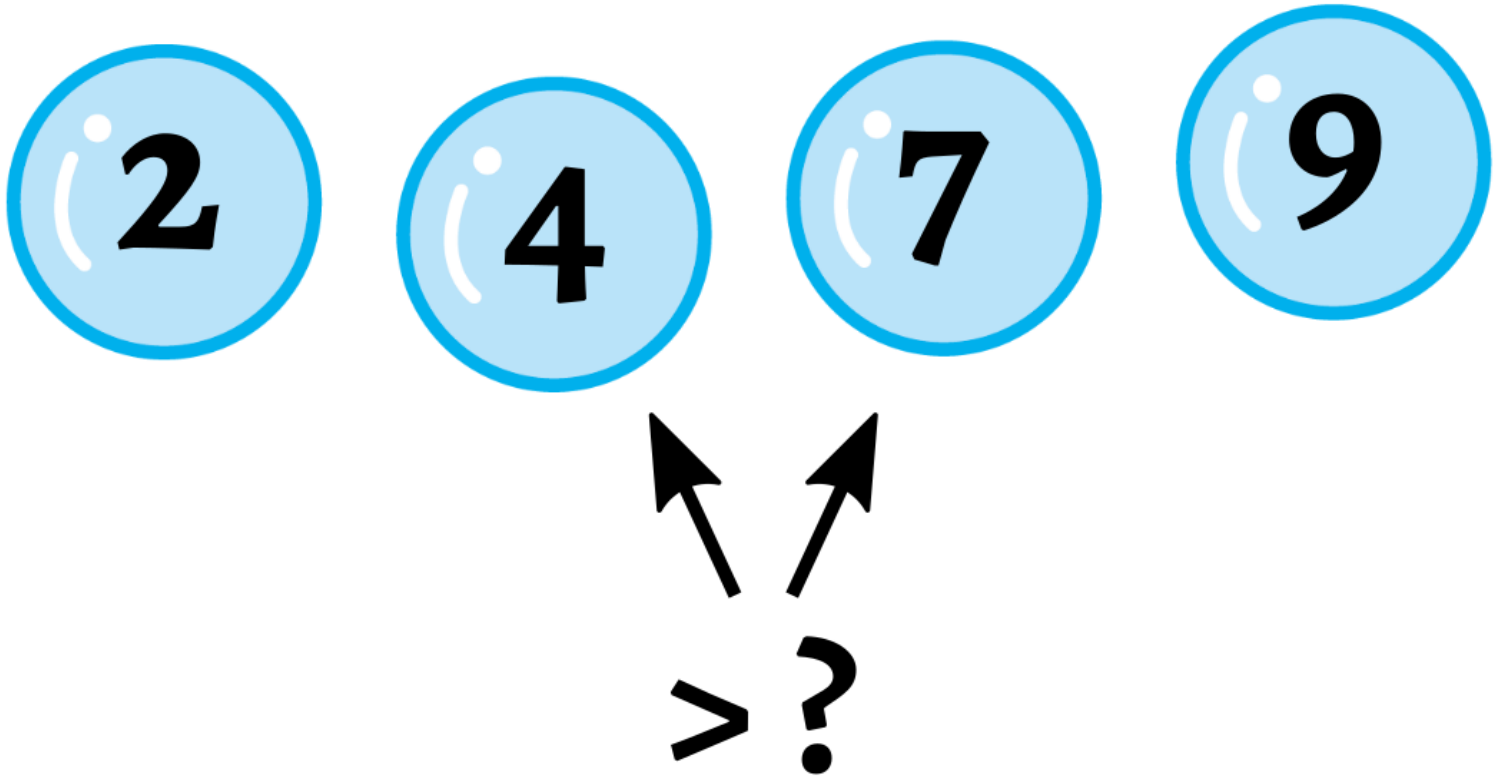
The Bubble Sort Algorithm



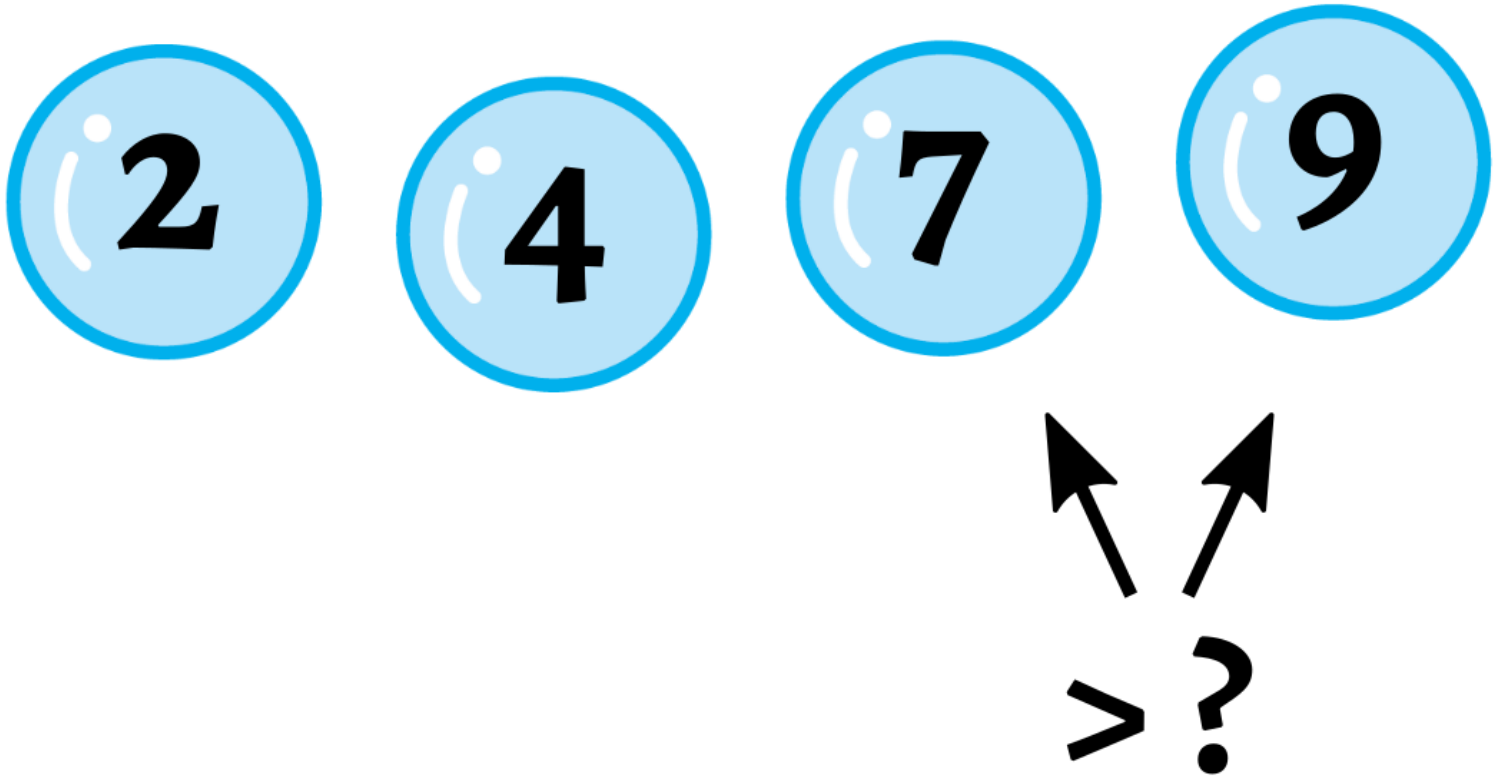
The Bubble Sort Algorithm



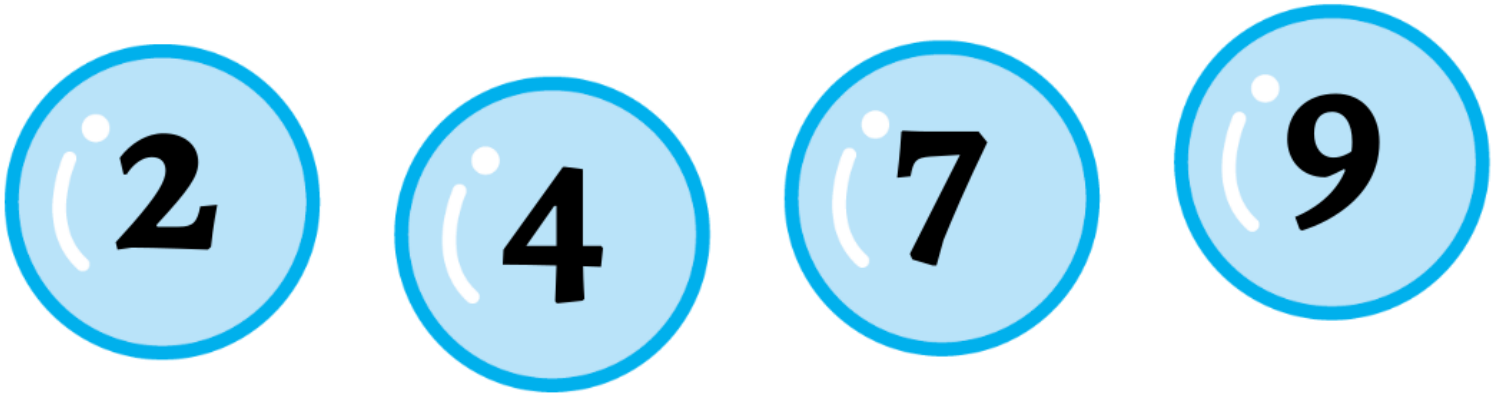
The Bubble Sort Algorithm



The Bubble Sort Algorithm



The Bubble Sort Algorithm



All pairs are in correct order!
Done!

Bubble Sort in Words

- Input: An unsorted list
- Do the following until nothing is changed anymore:
  - Iterate through the complete list
    1. Compare the current element with the next element
    2. If the current element is greater than the next element, switch their positions
    3. Notify whether a change was made
- The list is now sorted.

Helpful Functions

- The random module

```
import random #import the module similar to import math
#assigns dice_roll a number between 1 and 6
dice_roll = random.randint(1,6)
#random list item
myList = ["Rock","Paper","Scissors"]
random_item = myList[random.randint(0,len(myList)-1)]
```

- Convert a string to uppercase

```
name = "Peter"
upname = name.upper()
print(upname) # "PETER"
```

Task: Reverse a sentence

1. Write a script that reverts the word order in a given sentence
   - Let the user type in any sentence via the *input()* method
   - Split the sentence into a list of words
   - Use a for loop to go through the list in reverse order
   - During each iteration add the current word to a string variable *sentence*
   - Print the *sentence* variable

This is an example sentence $\rightarrow$ *sentence example an is This*

Task: Hangman

2. Write a Hangman computer game. The computer secretly chooses a word and the user may guess letters until the word is found.
   - Choose a random word from the words list and store it in variable
   - For each letter of the Word print an underscore "_"
   - Start a while loop that runs until the whole word is found
   - In the loop let the user guess a character and store the guessed character in a list
   - Run a second loop through each letter of the word and check whether this letter has been guessed already. If it has been guessed, print it otherwise print an underscore "_".
   - If you still had to replace a word by "_" the while loop continues

$$TASK \rightarrow \_\ \_\ \_\ \_$$

Task: Bubble Sort

3. Implement the Bubbling Sort Algorithm to sort a list of numbers
   - Start a while loop
   - In the while loop iterate through the list and compare the current and the next element
   - If the next element is smaller than the current one swap them
   - If you swap, make sure that the while loop is continued
   - If you did not swap at all, make sure the while loop ends