Foreign Student Visualization

Objects and constructors,
insertion sort algorithm

Making a plan:

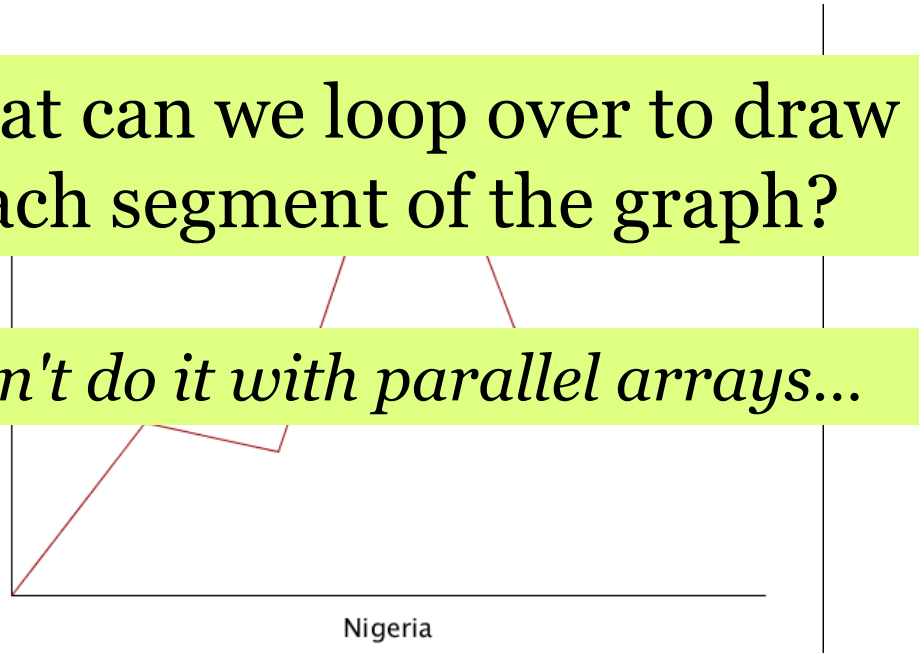
1. Create arrays to store data from the CSV file.
2. Read the data from the file and store it in the arrays.
3. Display the data flexibly so any number of countries can be shown.
4. Find what country (if any) was clicked on when the mouse is pressed.
5. Display a line graph when a country was clicked.
6. Change the sorting order when the s key is pressed.

Step 5

Display a line graph when a country was clicked.

What can we loop over to draw each segment of the graph?

Can't do it with parallel arrays...



We need a way to group all information about a line graph into one variable so we can loop over a single array of variables

Classes

Objects

Objects

Class Definition

Object Instance

Objects

Class Definition

Object Instance

Object Instance

Objects

Class Definition

Attributes
Behaviors

Object Instance

Attributes
Behaviors

Defining a Class

```
class Ball
{
   int x;
   int y;
   int size;
}
```

Defining a Class

```
class Ball
{
   int x;
   int y;
   int size;
}
```

Name of class: a new data type to make variables with

Defining a Class

```
class Ball
{
  int x;
  int y;
  int size;
}
```

Attributes: any variable type is allowed here

Defining a Class

```
class Ball
{
   int x;
   int y;
   int size;
}
```

This class has no behaviours

Creating and Modifying an Object

```
// Create an instance of Ball
Ball myNewBall = new Ball();

// Modify the attributes
myNewBall.x = width/2;
myNewBall.y = height/2;
myNewBall.size = 15;
```

Creating and Modifying an Object

Declare a new variable of type Ball

```
// Create an instance of Ball
Ball myNewBall = new Ball();

// Modify the attributes
myNewBall.x = width/2;
myNewBall.y = height/2;
myNewBall.size = 15;
```

Creating and Modifying an Object

Make a new object instance using the Ball class

```
// Create an instance
Ball myNewBall = new Ball();

// Modify the attributes
myNewBall.x = width/2;
myNewBall.y = height/2;
myNewBall.size = 15;
```

Creating and Modifying an Object

Modify the attributes defined in the class

```
instance of Ball
l = new Ball();

 attributes
myNewBall.x = width/2;
myNewBall.y = height/2;
myNewBall.size = 15;
```

Poll Everywhere Question

What is the output of the following code?

```
class Ball
{
  int x;
  int y;
  Ball b;
}

Ball b1 = new Ball();
b1.x = 10;
b1.y = 20;

Ball b2 = new Ball();
b2.x = 25;
b2.y = 45;

b2.b = b1;
b1.b = b2;
b2.x = b1.b.y;

println(b2.x);
```

Text 37607

334987: 10
653527: 20
653528: 25
663671: 45
663672: Nothing/ error

Constructors

Allow us to specify some of the attributes right when we make the object.

Defining a Class with a Constructor

```
class Ball
{
   int x;
   int y;
   int size;

   Ball(int x, int y, int size)
   {
      this.x = x;
      this.y = y;
      this.size = size;
   }
}
```

Defining a Class with a Constructor

```
class Ball
{
   int x;
   int y;
   int size;

   Ball(int x, int y, int size)
   {
      this.x = x;
      this.y = y;
      this.size = size;
   }
}
```

Like a function that will get called when object is created

Defining a Class with a Constructor

```
class Ball
{
   int x;
   int y;
   int size;

   Ball(int x, int y, int size)
   {
      this.x = x;
      this.y = y;
      this.size = size;
   }
}
```

Parameters usually used to set up initial values

Defining a Class with a Constructor

```
class Ball
{
   int x;
   int y;
   int size;

   Ball(int x, int
   {
      this.x = x;
      this.y = y;
      this.size = size;
   }
}
```

Saying 'this' allows us to refer to our own attributes

Creating an Object with a Constructor

```
// Create an instance of Ball
Ball myNewBall =
    new Ball(width/2, height/2, 15);
```

Creating an Object with a Constructor

```
// Create an instance of Ball
Ball myNewBall =
    new Ball(width/2, height/2, 15);
```

Now we create the object with the parameters we defined in the constructor

Object Design for Visualization

What we had before…

```
String[] countryNames;
int[]    numQ1_2013;
int[]    numQ2_2013;
int[]    numQ3_2013;
int[]    numQ4_2013;
int[]    numTotal_2013;
```

Object Design for Visualization

What we had before…

```
String[] countryNames;
int[]    numQ1_2013;
int[]    numQ2_2013;
int[]    numQ3_2013;
int[]    numQ4_2013;
int[]    numTotal_2013;
```

We couldn't loop through these for one country to draw the line graph

Object Design for Visualization

What we need now:

```
class ForeignStudentData
{
  String countryName;
  int[] quarters_2013;
  int total_2013;
}

ForeignStudentData[] countryData;
```

Object Design for Visualization

What we need now:

```
class ForeignStudentData
{
  String countryName;
  int[] quarters_2013;
  int total_2013;
}

ForeignStudentData[] cou
```

Now all quarterly data will be contained in a single country object and we can loop through it!

Creating Objects for Visualization

```
countryData[countryNum] = new ForeignStudentData();

countryData[countryNum].countryName = splitLine[0];

countryData[countryNum].quarters_2013 = new int[4];
countryData[countryNum].quarters_2013[0] = …;
countryData[countryNum].quarters_2013[1] = …;
countryData[countryNum].quarters_2013[2] = …;
countryData[countryNum].quarters_2013[3] = …;

countryData[countryNum].total_2013 = …;
```

Creating Objects for Visualization

```
countryData[countryNum] = new ForeignStudentData();

countryData[countryNum].co…;

countryData[countryNum].qu…;
countryData[countryNum].quarters_2013[0] = …;
countryData[countryNum].quarters_2013[1] = …;
countryData[countryNum].quarters_2013[2] = …;
countryData[countryNum].quarters_2013[3] = …;

countryData[countryNum].total_2013 = …;
```

Construct a new data object…

Creating Objects for Visualization

```
countryData[countryNum] = new ForeignStudentData();

c                        countryName = splitLine[0];

c                        quarters_2013 = new int[4];
c                        quarters_2013[0] = …;
c                        quarters_2013[1] = …;
c                        quarters_2013[2] = …;
countryData[countryNum].quarters_2013[3] = …;

countryData[countryNum].total_2013 = …;
```

…and assign the result into the `countryData` array

Creating Objects for Visualization

```
countryData[countryNum] = new ForeignStudentData();

countryData[countryNum].countryName = splitLine[0];

                       .quarters_2013 = new int[4];
                       .quarters_2013[0] = …;
                       .quarters_2013[1] = …;
                       .quarters_2013[2] = …;
                       .quarters_2013[3] = …;

countryData[countryNum].total_2013 = …;
```

The object is stored at this slot, so this part of the code retrieves it for us

Creating Objects for Visualization

```
countryData[countryNum] = new ForeignStudentData();

countryData[countryNum].countryName = splitLine[0];

countryData[count                                  ew int[4];
countryData[count                                  = …;
countryData[count                                  = …;
countryData[count                                  = …;
countryData[countryNum].quarters_2013[3] = …;

countryData[countryNum].total_2013 = …;
```

Once we have the object, we can access its attributes

Creating Objects for Visualization

```
countryData[countryNum] = new ForeignStudentData();

countryData[countryNum].countryName = splitLine[0];

countryData[countryNum].quarters_2013 = new int[4];
countryData[countryNum
countryData[countryNum
countryData[countryNum
countryData[countryNum

countryData[countryNum
```

We have to initialize all arrays before we use them, even those inside an object

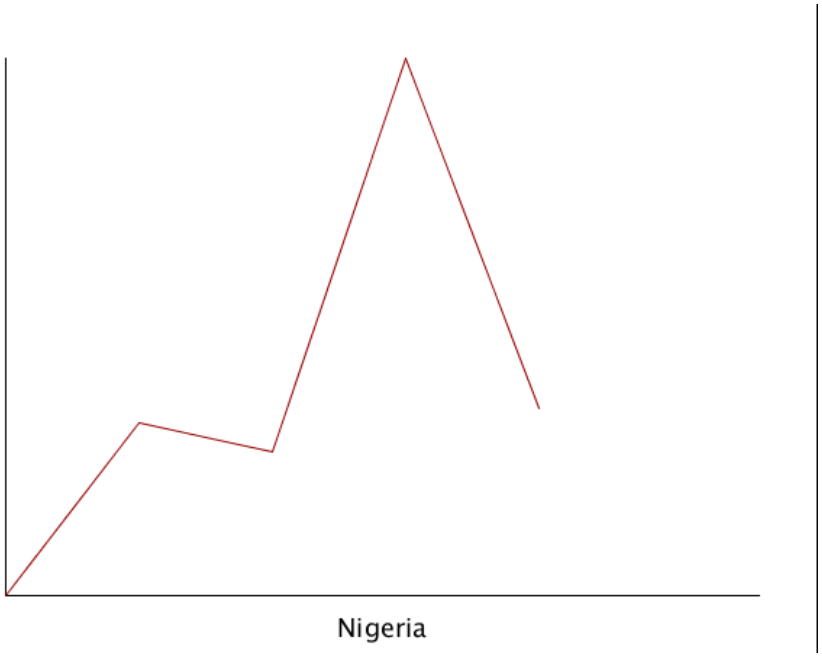
Creating Objects for Visualization

```
countryData[countryNum] = new ForeignStudentData();

countryData[countryNum].countryName = splitLine[0];

countryData[countryNum].quarters_2013 = new int[4];
countryData[countryNum].quarters_2013[0] = …;
countryData[countryNum].quarters_2013[1] = …;
countryData[countryNum].quarters_2013[2] = …;
countryData[countryNum].quarters_2013[3] = …;

countryData[countryNum]
```

Once initialized, we can start assigning values to the array inside the object

Step 5

Display a line graph when a country was clicked.



Drawing the Line Graph

```
float lastPointX = axisX;
float lastPointY = axisY;

int pointNum = 0;
while (pointNum < numPoints)
{
  float yValue = countryData[index].quarters_2013[pointNum];
  yValue = yValue / maxValue * axisHeight;
  yValue = axisY - yValue;

  line(lastPointX, lastPointY, lastPointX+spacing, yValue);
  lastPointX = lastPointX+spacing;
  lastPointY = yValue;
  pointNum++;
}
```

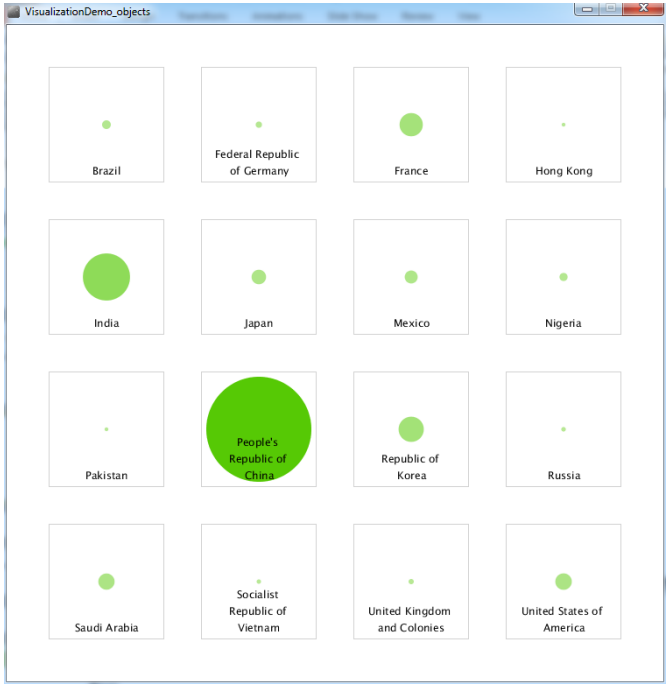
Drawing the Line Graph

Now we have a single array with quarterly data that we can loop through

```
float lastPointX = axisX;
float lastPointY = axisY;

int pointNum = 0;
while (pointNum < numPoints)
{
  float yValue = countryData[index].quarters_2013[pointNum];
  yValue = yValue / maxValue * axisHeight;
  yValue = axisY - yValue;

  line(lastPointX, lastPointY, lastPointX+spacing, yValue);
  lastPointX = lastPointX+spacing;
  lastPointY = yValue;
  pointNum++;
}
```

Step 6

Change the sorting order when the s key is pressed.



Insertion Sort Algorithm

High level:

Take next item from unsorted list, insert it into proper place in the sorted list.

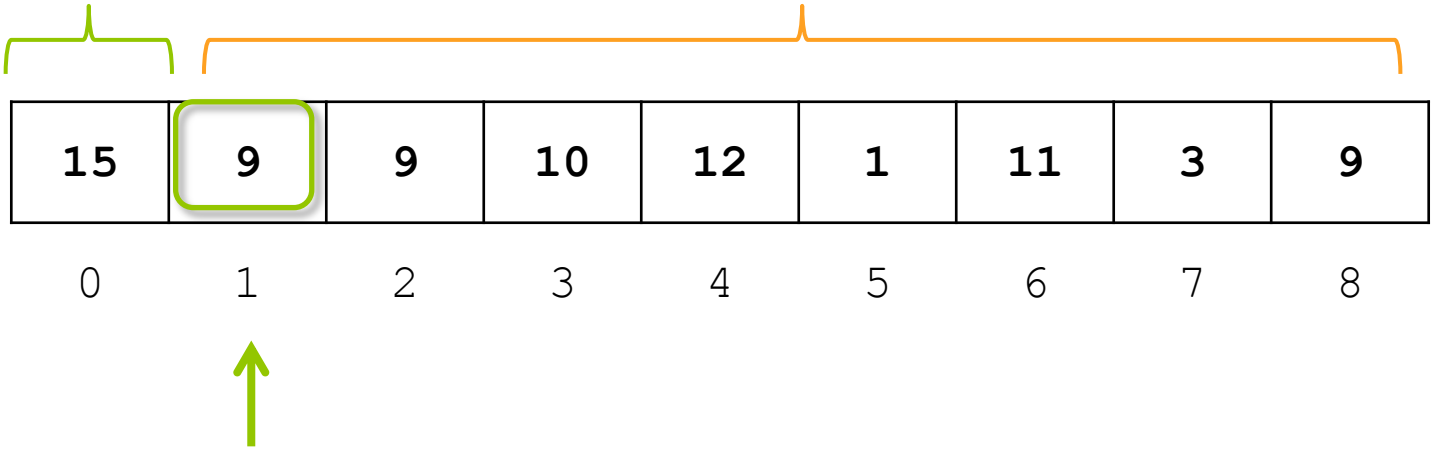
Insertion Sort Algorithm

http://algo-rythmics.ms.sapientia.ro/dance/insertion

Insertion Sort

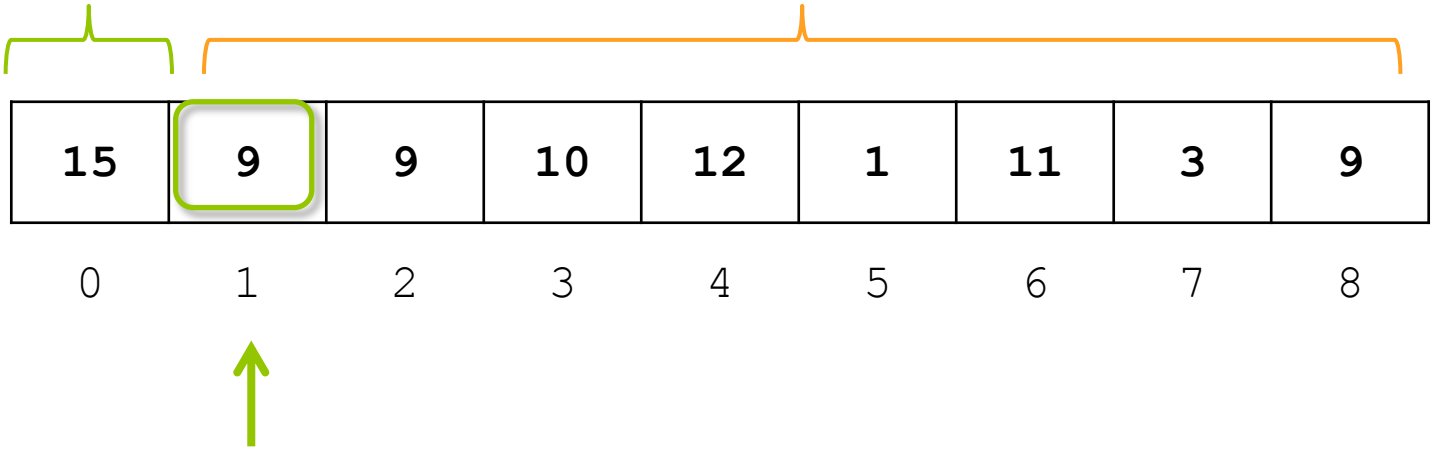
```
input: a list of data to sort
output: the input list will be sorted

set currentStartIndex to 1

while currentStartIndex < length of list:

    set innerIndex to currentStartIndex

    while innerIndex > 0 AND
          value at innerIndex < value at (innerIndex-1):

        swap values as innerIndex and (innerIndex-1)
        decrease innerIndex by 1

    increase currentStartIndex by 1
```

Insertion Sort Algorithm



currentStartIndex: 1
innerIndex: 1

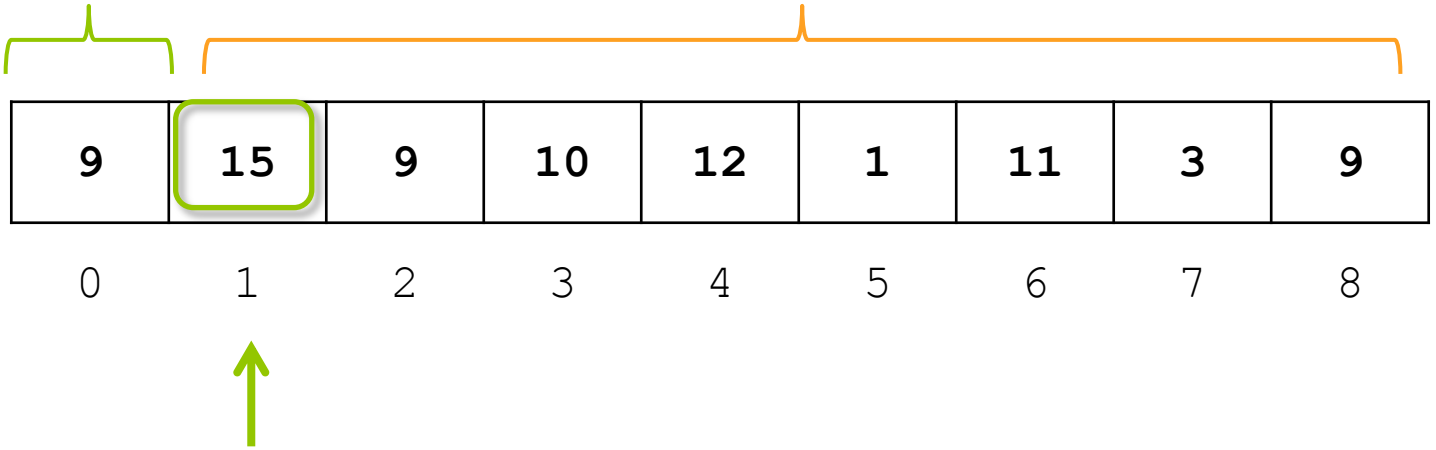
Insertion Sort Algorithm



9 smaller than 15,
so they need to swap

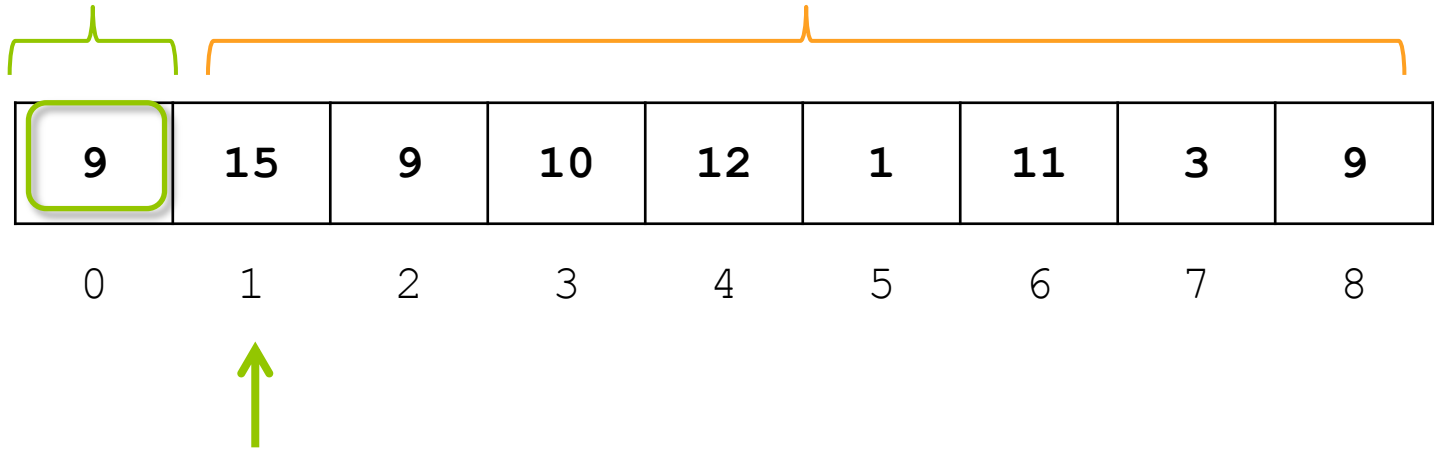
currentStartIndex: 1
innerIndex: 1

Insertion Sort Algorithm



currentStartIndex: 1
innerIndex: 1

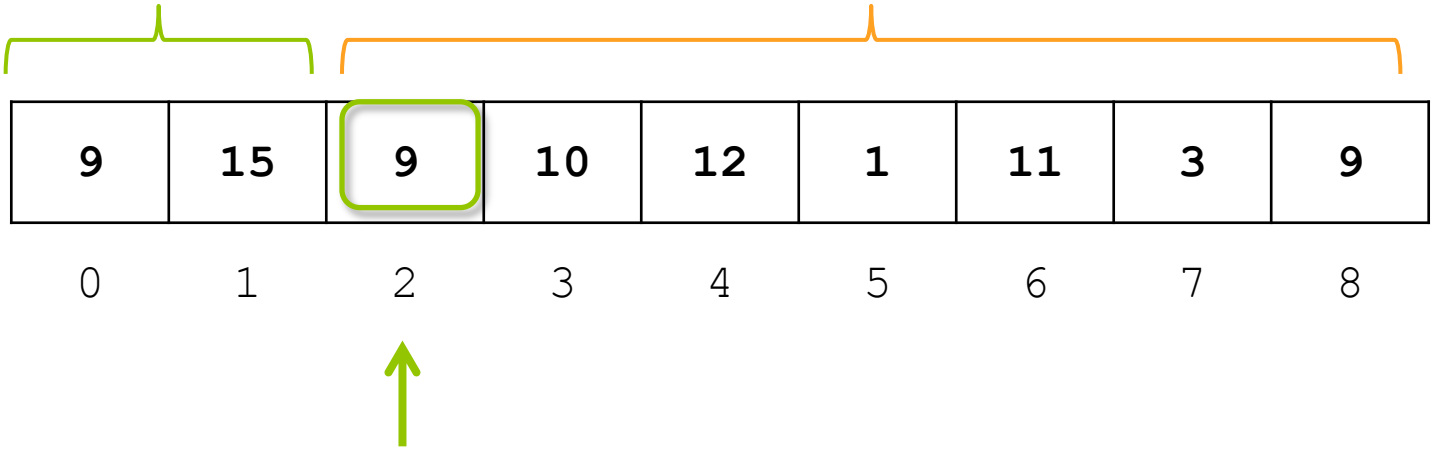
Insertion Sort Algorithm



innerIndex no longer bigger than zero, inner loop exits

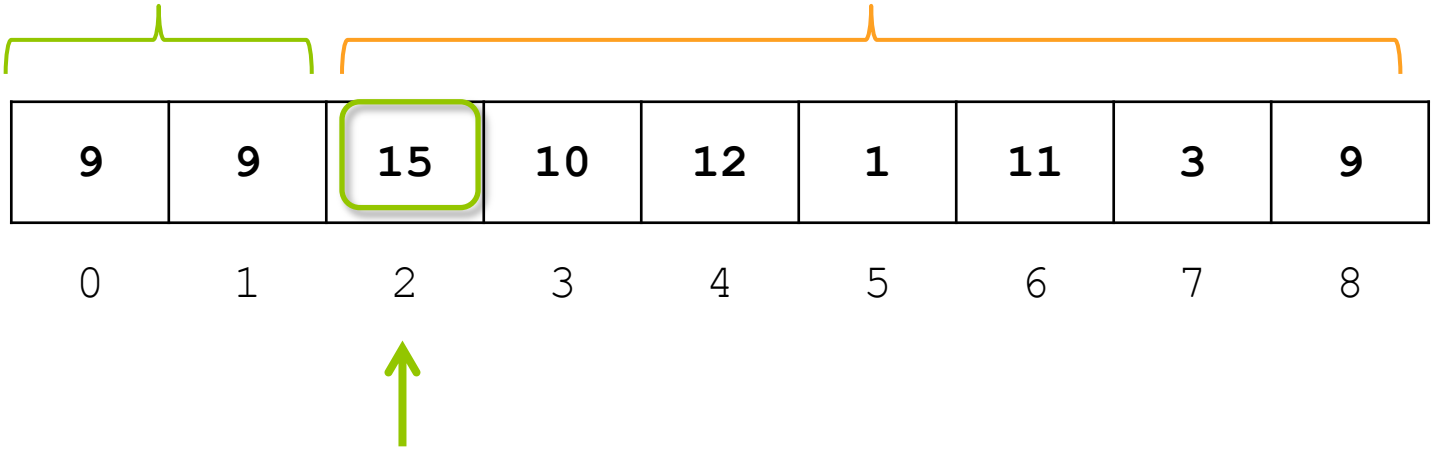
currentStartIndex: 1
innerIndex: 0

Insertion Sort Algorithm



currentStartIndex: 2
innerIndex: 2

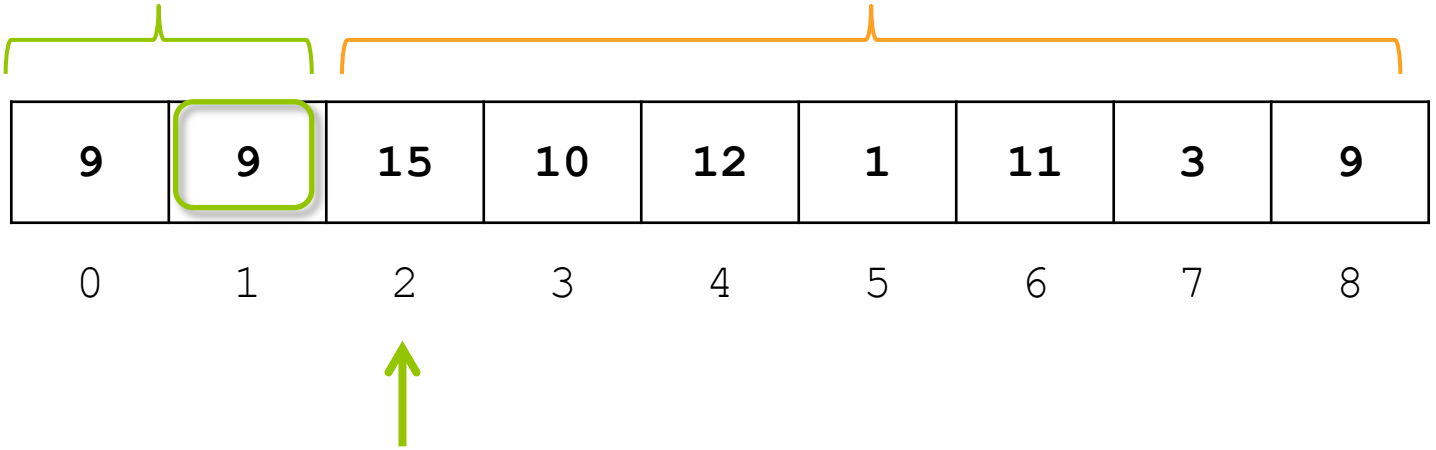
Insertion Sort Algorithm



9 and 15 swapped

currentStartIndex: 2
innerIndex: 2

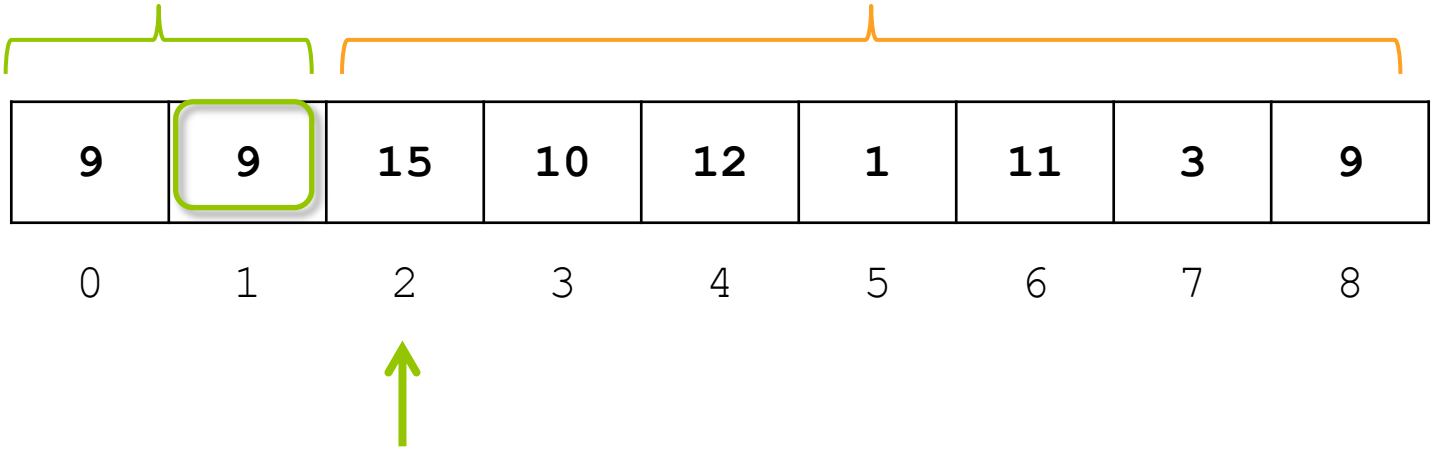
Insertion Sort Algorithm



innerIndex moved left

currentStartIndex: 2
innerIndex: 1

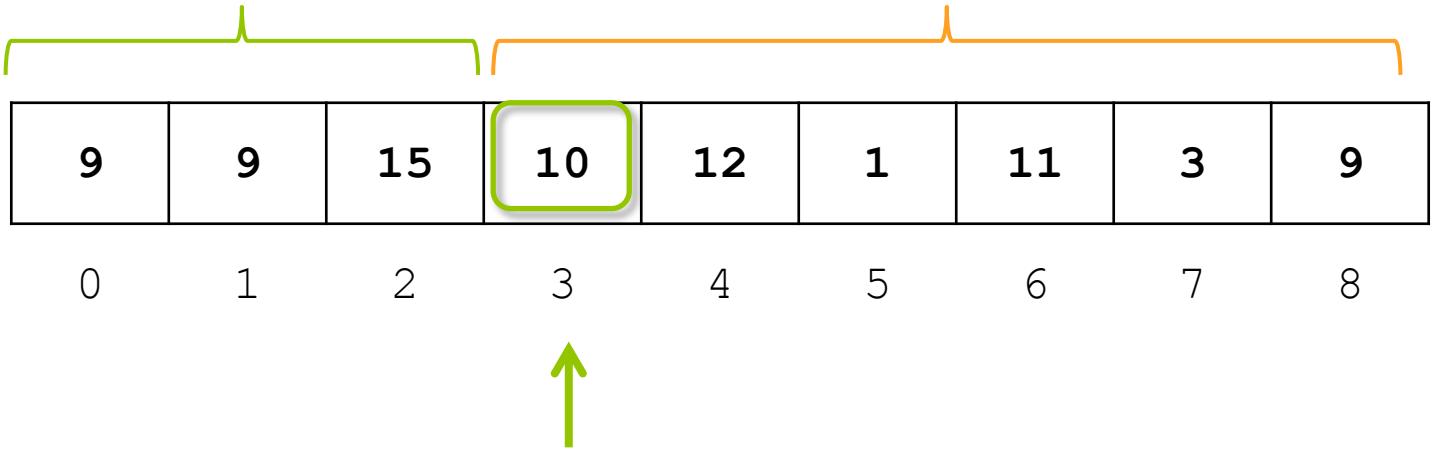
Insertion Sort Algorithm



9 is not smaller than 9,
inner loops exits

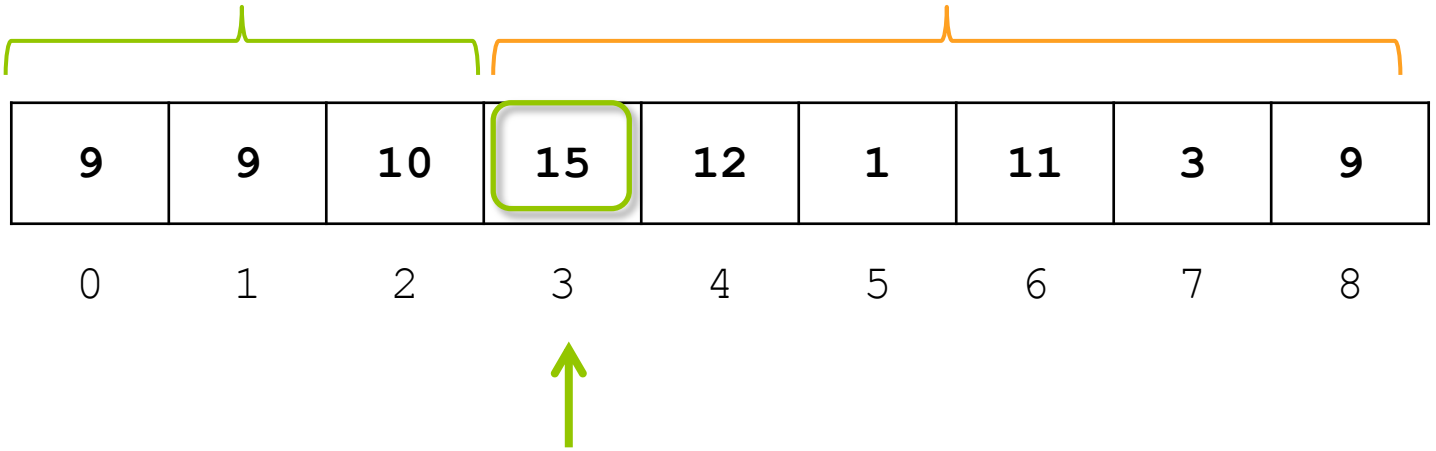
currentStartIndex: 2
innerIndex: 1

Insertion Sort Algorithm



currentStartIndex: 3
innerIndex: 3

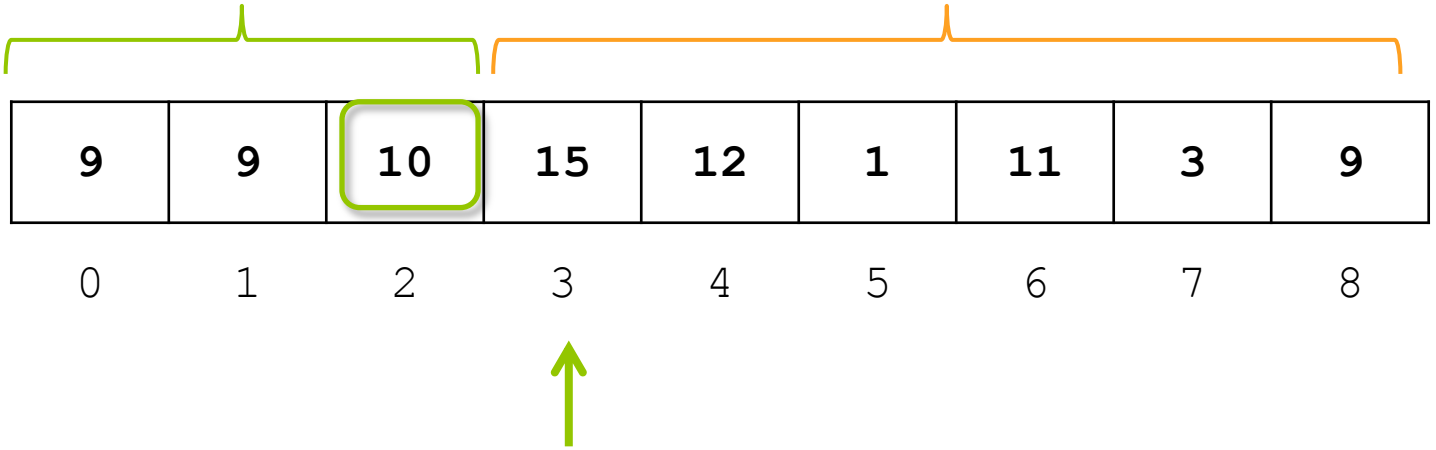
Insertion Sort Algorithm



10 and 15 swapped

currentStartIndex: 3
innerIndex: 3

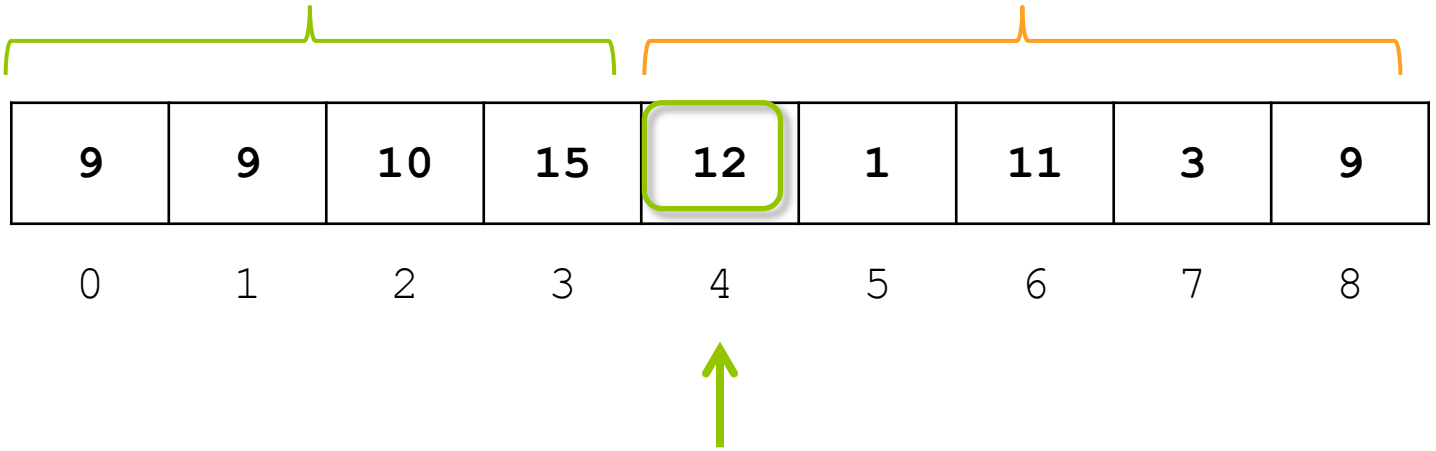
Insertion Sort Algorithm



10 is not smaller than 9,
inner loops exits

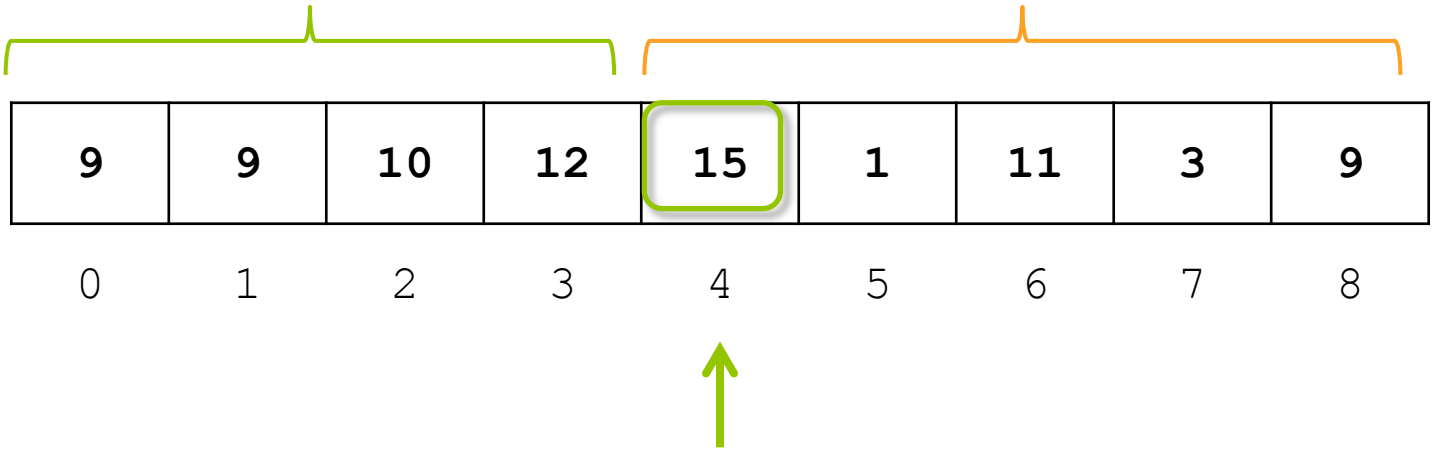
currentStartIndex: 3
innerIndex: 2

Insertion Sort Algorithm



currentStartIndex: 4
innerIndex: 4

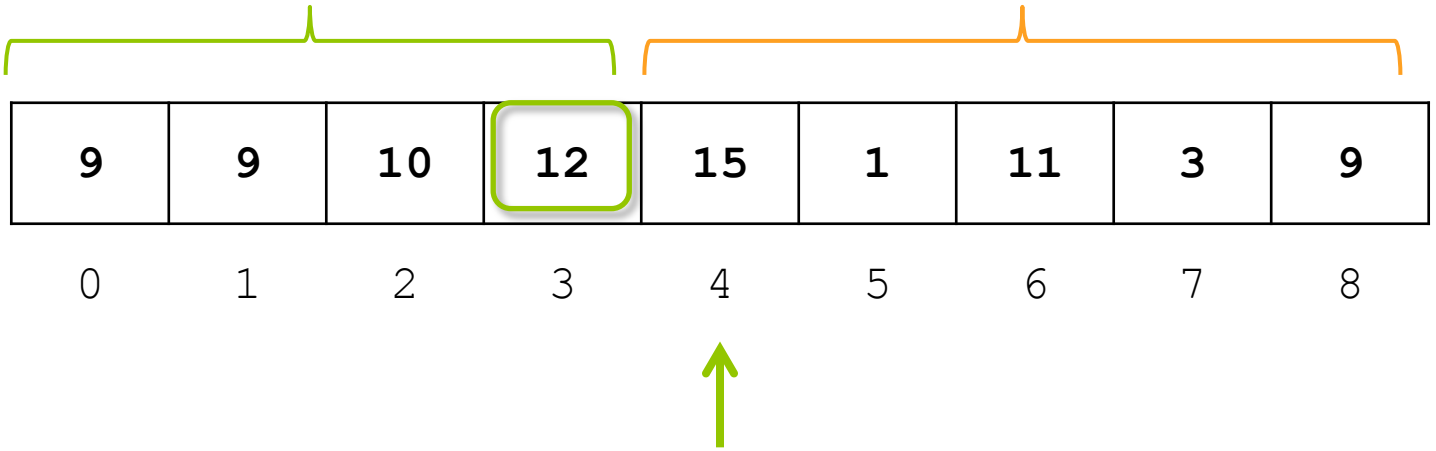
Insertion Sort Algorithm



12 and 15 swap

currentStartIndex: 4
innerIndex: 4

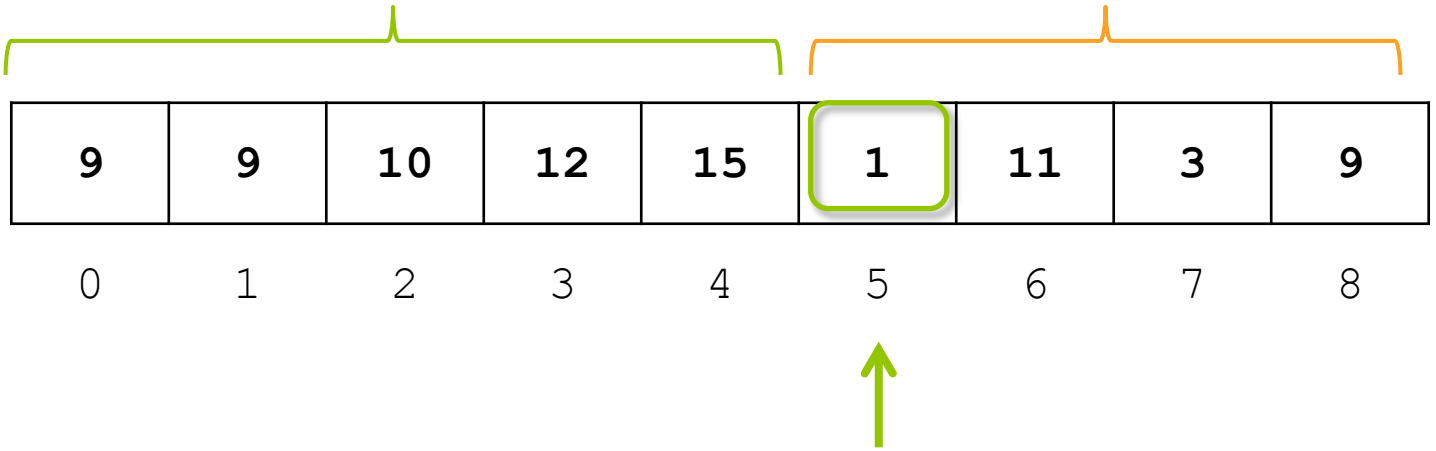
Insertion Sort Algorithm



12 is not smaller than 10,
inner loops exits

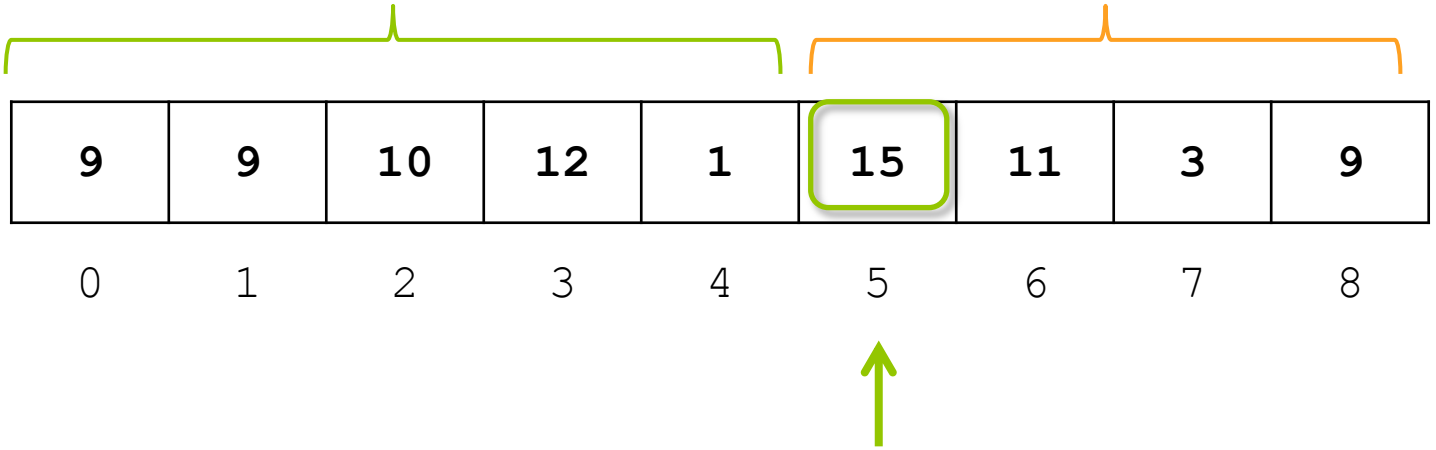
currentStartIndex: 4
innerIndex: 3

Insertion Sort Algorithm



currentStartIndex: 5
innerIndex: 5

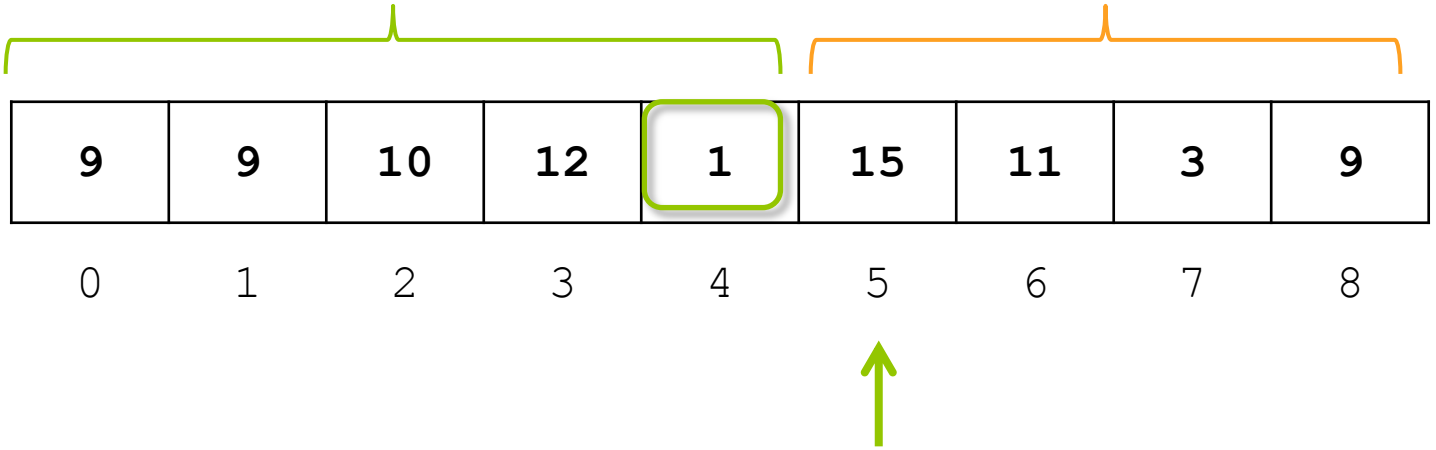
Insertion Sort Algorithm



1 and 15 swap

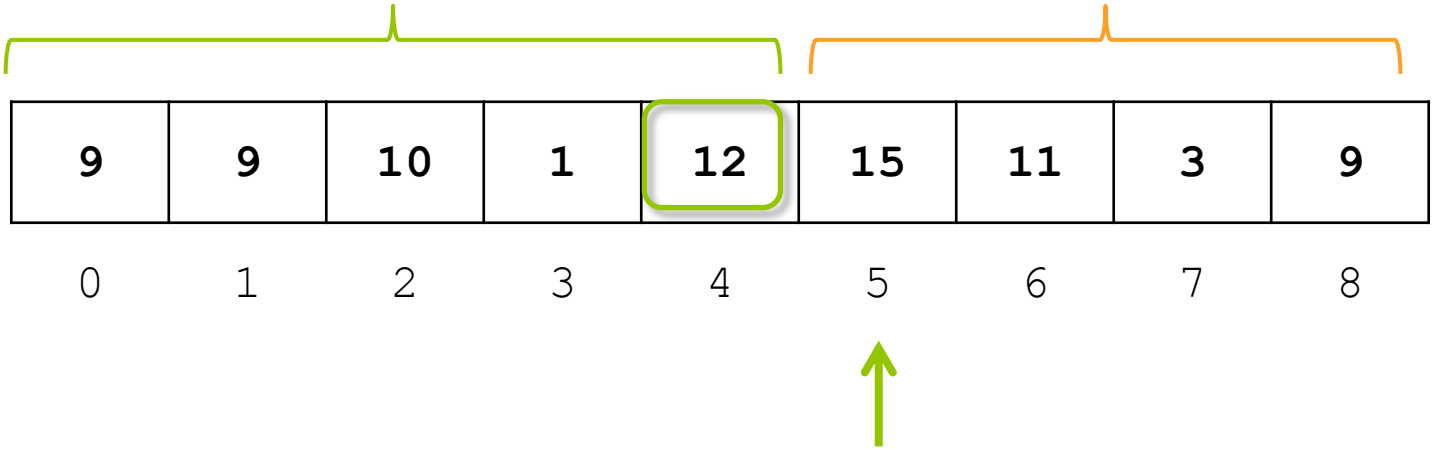
currentStartIndex: 5
innerIndex: 5

Insertion Sort Algorithm



currentStartIndex: 5
innerIndex: 4

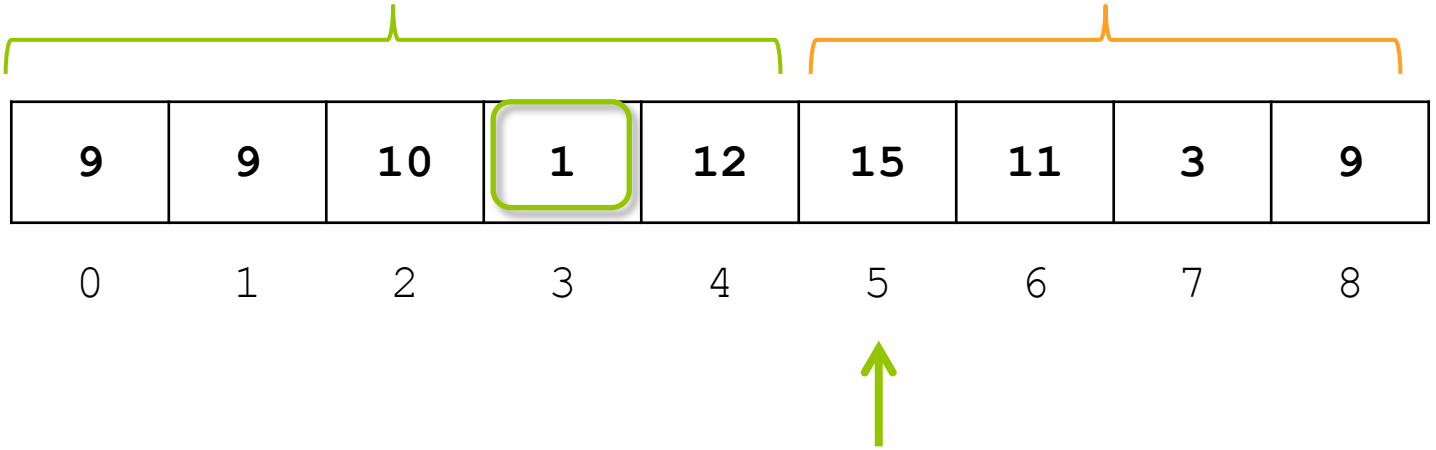
Insertion Sort Algorithm



1 and 12 swap

currentStartIndex: 5
innerIndex: 4

Insertion Sort Algorithm

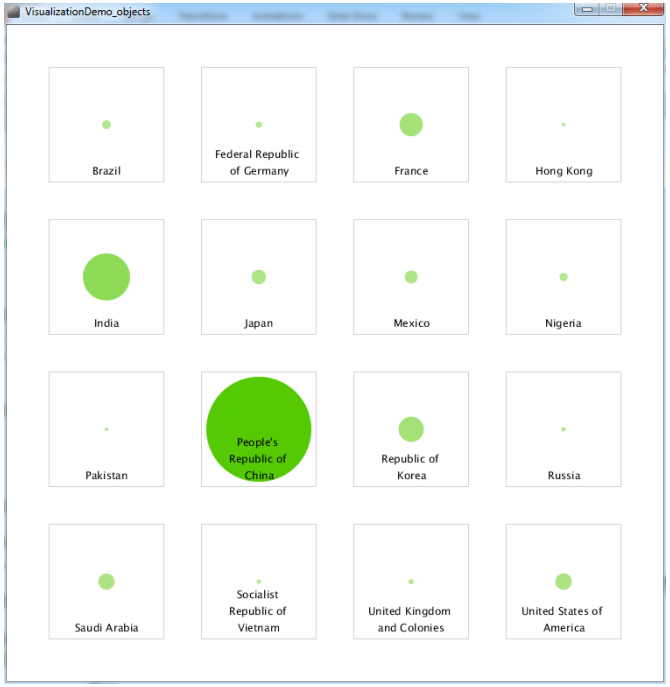


...etc...

currentStartIndex: 5
innerIndex: 3

Step 6

Change the sorting order when the s key is pressed.



Sorting for Visualization

We need two different sorted orders:

1. Total number of students in 2013
2. Alphabetical according to country name

Sorting for Visualization

We need two different sorted orders:

1. Total number of students in 2013
2. Alphabetical according to country name

All that changes is the check for relative size in the inner loop!