

C307 Data Representation

Unit-2, Part B_5 Array of Search and Sort

By:



Dr. Hossein Hakimzadeh
Computer Science and Informatics
IU South Bend

Additional Array Operations

- Search
 - Linear vs. Binary
- Sort
- Other operations:
 - Min
 - Max
 - Mean
 - Median
 - Mode
 - Stddev

	Values	Sorted Values
	77	-1
	99	0
	44	11
	55	11
	-1	33
	88	44
	11	55
	0	55
	66	66
	33	77
	11	88
	55	99
Mean	44.83333333	44.83333333
Median	49.5	49.5
Mode	55	11
Stddev	34.41414438	34.41414438
Min	-1	-1
Max	99	99

StdDev

$$s = \sqrt{\frac{\sum_{i=1}^N (x_i - \bar{x})^2}{N - 1}}$$

https://en.wikipedia.org/wiki/Standard_deviation

- is a measure that is used to quantify the amount of variation or dispersion of a set of data values.
- A low standard deviation indicates that the data points tend to be close to the mean.
- a high standard deviation indicates that the data points are spread out over a wider range of values.

Mode

- ❑ The mode of a set of data values is the value that appears most often.
- ❑ The mode is not necessarily unique to a given discrete distribution, since the probability mass function may take the same maximum value at several points x_1, x_2 , etc. The most extreme case occurs in uniform distributions, where all values occur equally frequently.

	Values	Sorted Values
	77	-1
	99	0
	44	11
	55	11
	-1	33
	88	44
	11	55
	0	55
	66	66
	33	77
	11	88
	55	99
Mean	44.83333333	44.83333333
Median	49.5	49.5
Mode	55	11
Stddev	34.41414438	34.41414438
Min	-1	-1
Max	99	99

Median

- The median is the value separating the higher half from the lower half of a data sample

1, 3, 3, **6**, 7, 8, 9

Median = **6**

1, 2, 3, **4**, **5**, 6, 8, 9

Median = $(4 + 5) \div 2$
= **4.5**

<https://en.wikipedia.org/wiki/Median>

```

public static void main(String[] args) {
    int maxSize = 100;        // array size
    HHArray_V2 myArray;        // reference to array
    myArray = new HHArray_V2(maxSize); // create the array

    System.out.println("-----");
    System.out.println("- Insert 10 numbers in the array--");
    myArray.insert(77);        // insert 10 items
    myArray.insert(99);
    myArray.insert(44);
    myArray.insert(55);
    myArray.insert(-1);
    myArray.insert(88);
    myArray.insert(11);
    myArray.insert(0);
    myArray.insert(66);
    myArray.insert(33);
    myArray.insert(11); // added to test MODE()
    myArray.insert(55); // added to test MODE()

    System.out.println("-----");
    System.out.println("----- Print Array -----");
    myArray.display();

    System.out.println("-----");
    System.out.println("----- Linear Search -----");
    int searchKey = 33;        // search for item
    int location = myArray.linearSearch(searchKey);
    if (location == -1) {
        System.out.println("Can't find " + searchKey);
    } else {
        System.out.println("Found " + searchKey + " at location: " + location);
    }

    System.out.println("-----");
    System.out.println("----- Binary Search -----");
    location = myArray.binarySearch(searchKey);
    if (location == -1) {
        System.out.println("Can't find " + searchKey);
    } else {
        System.out.println("Found " + searchKey + " at location: " + location);
    }
}

```

```

    System.out.println("-----");
    System.out.println("----- Bubble Sort -----");
    myArray.bubbleSort();

    System.out.println("-----");
    System.out.println("----- Print Array -----");
    myArray.display();

    System.out.println("-----");
    System.out.println("----- Binary Search -----");
    searchKey = 33;
    location = myArray.binarySearch(searchKey);
    if (location == -1) {
        System.out.println("Can't find " + searchKey);
    } else {
        System.out.println("Found " + searchKey + " at location: " + location);
    }

    System.out.println("-----");
    System.out.println("----- Min -----");
    System.out.println("Min = " + myArray.min());

    System.out.println("-----");
    System.out.println("----- Max -----");
    System.out.println("Max = " + myArray.max());

    System.out.println("-----");
    System.out.println("----- Mean -----");
    System.out.println("Mean = " + myArray.mean());

    System.out.println("-----");
    System.out.println("----- StdDev -----");
    System.out.println("StdDev = " + myArray.stdDev());

    System.out.println("-----");
    System.out.println("----- Mode -----");
    System.out.println("Mode = " + myArray.mode());
}

```

Binary Search

```
public int binarySearch(int searchKey) { // displays array contents

    int Low, High, Middle;
    Low = 0;
    High = nElems;

    while (Low <= High) {
        Middle = (int) ((Low + High) / 2);

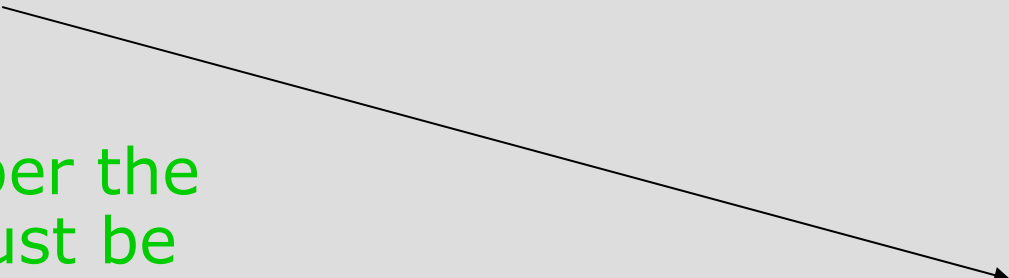
        if (searchKey == theArray[Middle]) { //Its a match!!
            return (Middle);
        } else {
            if (searchKey < theArray[Middle]) //'Search the low end of the array
            {
                High = Middle - 1;
            } else {
                Low = Middle + 1;
            }
        }
    }
    return (-1); //return -1 to indicate the value was not found
}
```

Consider the following Array:

- ❑ Binary Search of the array:

BinarySearch(17)

- Remember the array must be sorted first.



2	5	9	12	17	25	44	51	57	62	95
0	1	2	3	4	5	6	7	8	9	10

Binary Search:

```
Private Function BinarySearch(ByVal value As Integer, ByRef TheArray() As Integer) As Integer
```

```
    Dim Low, High, Middle As Integer
```

```
    Low = 0
```

```
    High = TheArray.GetUpperBound(0)
```

```
    Do While Low <= High
```

```
        Middle = (Low + High) \ 2
```

```
        If value = TheArray(Middle) Then 'It's a match!!
```

```
            Return (Middle)
```

```
        ElseIf value < TheArray(Middle) Then 'Search the low end of array
```

```
            High = Middle - 1
```

```
        Else
```

```
            Low = Middle + 1
```

```
        End If
```

```
    Loop
```

```
    Return (-1) 'return -1 to indicate the value was not found
```

```
End Function
```

2	5	9	12	17	25	44	51	57	62	95
0	1	2	3	4	5	6	7	8	9	10

Low

Middle

High

Binary Search:

```
Private Function BinarySearch(ByVal value As Integer, ByRef TheArray() As Integer) As Integer
```

```
    Dim Low, High, Middle As Integer
```

```
    Low = 0
```

```
    High = TheArray.GetUpperBound(0)
```

```
    Do While Low <= High
```

```
        Middle = (Low + High) \ 2
```

```
        If value = TheArray(Middle) Then 'It's a match!!
```

```
            Return (Middle)
```

```
        ElseIf value < TheArray(Middle) Then 'Search the low end of array
```

```
            High = Middle - 1
```

```
        Else
```

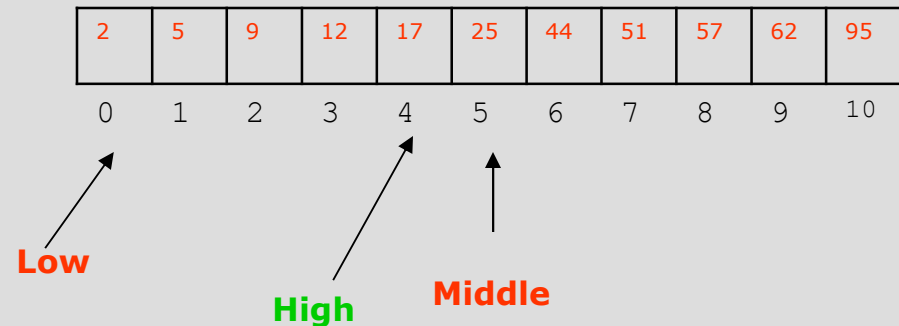
```
            Low = Middle + 1
```

```
        End If
```

```
    Loop
```

```
    Return (-1) 'return -1 to indicate the value was not found
```

```
End Function
```



Binary Search:

```
Private Function BinarySearch(ByVal value As Integer, ByRef TheArray() As Integer) As Integer
```

```
    Dim Low, High, Middle As Integer
```

```
    Low = 0
```

```
    High = TheArray.GetUpperBound(0)
```

```
    Do While Low <= High
```

```
        Middle = (Low + High) \ 2
```

```
        If value = TheArray(Middle) Then 'It's a match!!
```

```
            Return (Middle)
```

```
        ElseIf value < TheArray(Middle) Then 'Search the low end of array
```

```
            High = Middle - 1
```

```
        Else
```

```
            Low = Middle + 1
```

```
        End If
```

```
    Loop
```

```
    Return (-1) 'return -1 to indicate the value was not found
```

```
End Function
```



2	5	9	12	17	25	44	51	57	62	95
---	---	---	----	----	----	----	----	----	----	----

0 1 2 3 4 5 6 7 8 9 10

Low

High

Middle

Binary Search:

```
Private Function BinarySearch(ByVal value As Integer, ByRef TheArray() As Integer) As Integer
```

```
    Dim Low, High, Middle As Integer
```

```
    Low = 0
```

```
    High = TheArray.GetUpperBound(0)
```

```
    Do While Low <= High
```

```
        Middle = (Low + High) \ 2
```

```
        If value = TheArray(Middle) Then 'It's a match!!
```

```
            Return (Middle)
```

```
        ElseIf value < TheArray(Middle) Then 'Search the low end of array
```

```
            High = Middle - 1
```

```
        Else
```

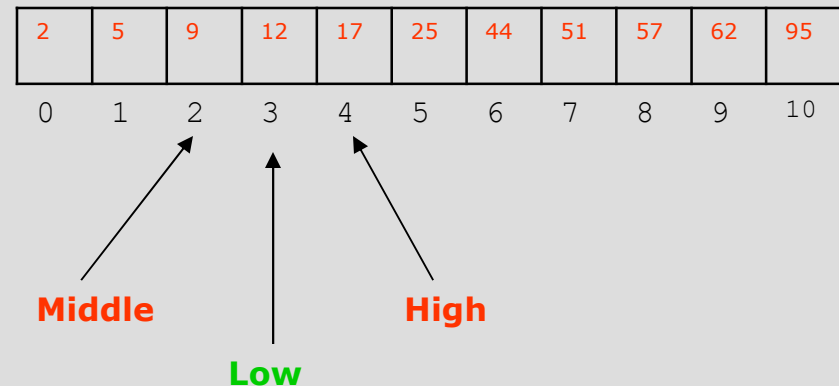
```
            Low = Middle + 1
```

```
        End If
```

```
    Loop
```

```
    Return (-1) 'return -1 to indicate the value was not found
```

```
End Function
```



Binary Search:

```
Private Function BinarySearch(ByVal value As Integer, ByRef TheArray() As Integer) As Integer
    Dim Low, High, Middle As Integer
    Low = 0
    High = TheArray.GetUpperBound(0)
```

```
    Do While Low <= High
```

```
        Middle = (Low + High) \ 2
```

```
        If value = TheArray(Middle) Then 'It's a match!!
```

```
            Return (Middle)
```

```
        ElseIf value < TheArray(Middle) Then 'Search the low end of array
```

```
            High = Middle - 1
```

```
        Else
```

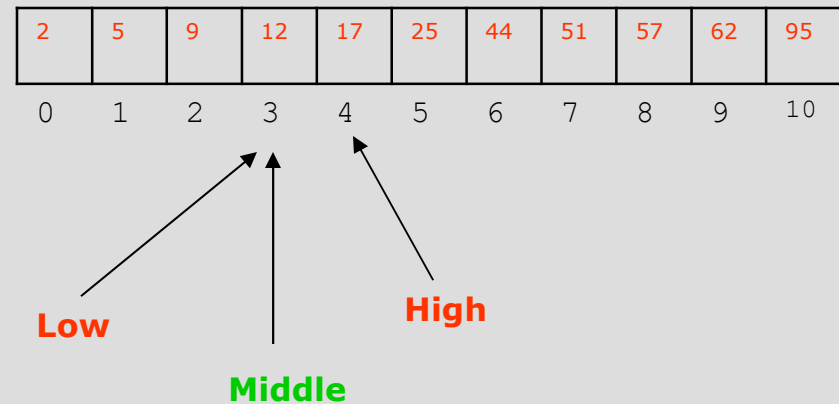
```
            Low = Middle + 1
```

```
        End If
```

```
    Loop
```

```
    Return (-1) 'return -1 to indicate the value was not found
```

```
End Function
```



Binary Search:

```
Private Function BinarySearch(ByVal value As Integer, ByRef TheArray() As Integer) As Integer
```

```
    Dim Low, High, Middle As Integer
```

```
    Low = 0
```

```
    High = TheArray.GetUpperBound(0)
```

```
    Do While Low <= High
```

```
        Middle = (Low + High) \ 2
```

```
        If value = TheArray(Middle) Then 'It's a match!!
```

```
            Return (Middle)
```

```
        ElseIf value < TheArray(Middle) Then 'Search the low end of array
```

```
            High = Middle - 1
```

```
        Else
```

```
            Low = Middle + 1
```

```
        End If
```

```
    Loop
```

```
    Return (-1) 'return -1 to indicate the value was not found
```

```
End Function
```

2	5	9	12	17	25	44	51	57	62	95
0	1	2	3	4	5	6	7	8	9	10

Middle

Low

High

Binary Search:

```
Private Function BinarySearch(ByVal value As Integer, ByRef TheArray() As Integer) As Integer
```

```
    Dim Low, High, Middle As Integer
```

```
    Low = 0
```

```
    High = TheArray.GetUpperBound(0)
```

```
    Do While Low <= High
```

```
        Middle = (Low + High) \ 2
```

```
        If value = TheArray(Middle) Then 'It's a match!!
```

```
            Return (Middle)
```

```
        ElseIf value < TheArray(Middle) Then 'Search the low end of array
```

```
            High = Middle - 1
```

```
        Else
```

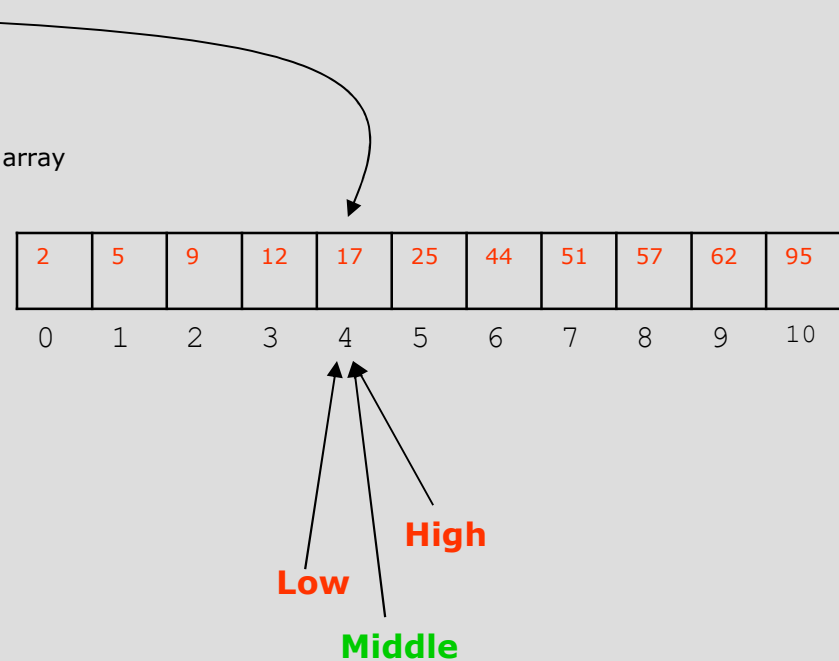
```
            Low = Middle + 1
```

```
        End If
```

```
    Loop
```

```
    Return (-1) 'return -1 to indicate the value was not found
```

```
End Function
```



2	5	9	12	17	25	44	51	57	62	95
0	1	2	3	4	5	6	7	8	9	10

Low

High

Middle

Binary Search:

```
Private Function BinarySearch(ByVal value As Integer, ByRef TheArray() As Integer) As Integer
```

```
    Dim Low, High, Middle As Integer
```

```
    Low = 0
```

```
    High = TheArray.GetUpperBound(0)
```

```
    Do While Low <= High
```

```
        Middle = (Low + High) \ 2
```

```
        If value = TheArray(Middle) Then 'It's a match!!
```

```
            Return (Middle)
```

```
        ElseIf value < TheArray(Middle) Then 'Search the low end of array
```

```
            High = Middle - 1
```

```
        Else
```

```
            Low = Middle + 1
```

```
        End If
```

```
    Loop
```

```
    Return (-1) 'return -1 to indicate the value was not found
```

```
End Function
```



2	5	9	12	17	25	44	51	57	62	95
0	1	2	3	4	5	6	7	8	9	10

Low

High

Middle

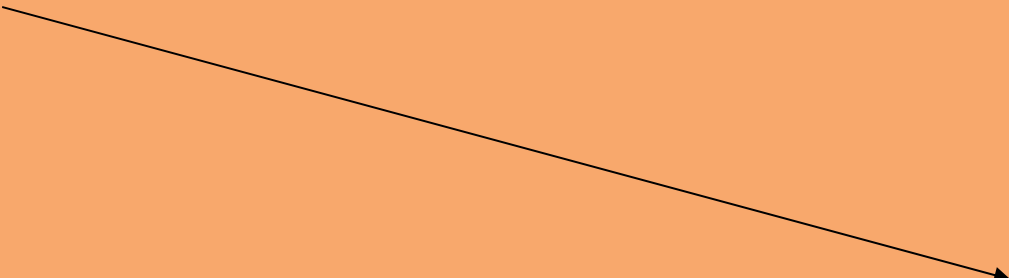
Bubble Sort()

```
public void bubbleSort() {  
    int pass, index, hold;  
  
    for (pass = 1; pass < nElems; pass++) {  
        for (index = 0; index < nElems - 1; index++) {  
            if (theArray[index] > theArray[index + 1]) {  
                //swap  
                hold = theArray[index];  
                theArray[index] = theArray[index + 1];  
                theArray[index + 1] = hold;  
            }  
        }  
    }  
}
```

Consider the following Array:

- Linear Search of the array:

BubbleSort(TheArray)



2	51	12	5	9	44	57	25	17	62	95
0	1	2	3	4	5	6	7	8	9	10

Bubble Sort:

```
Private Sub BubbleSort(ByRef TheArray() As Integer)
```

```
    Dim Pass, Index, Hold As Integer
```

Hold

```
    For Pass = 1 To TheArray.GetUpperBound(0)
```

```
        For Index = 0 To TheArray.GetUpperBound(0) - 1
```

```
            If TheArray(Index) > TheArray(Index + 1) Then
```

```
                Hold = TheArray(Index)
```

```
                TheArray(Index) = TheArray(Index + 1)
```

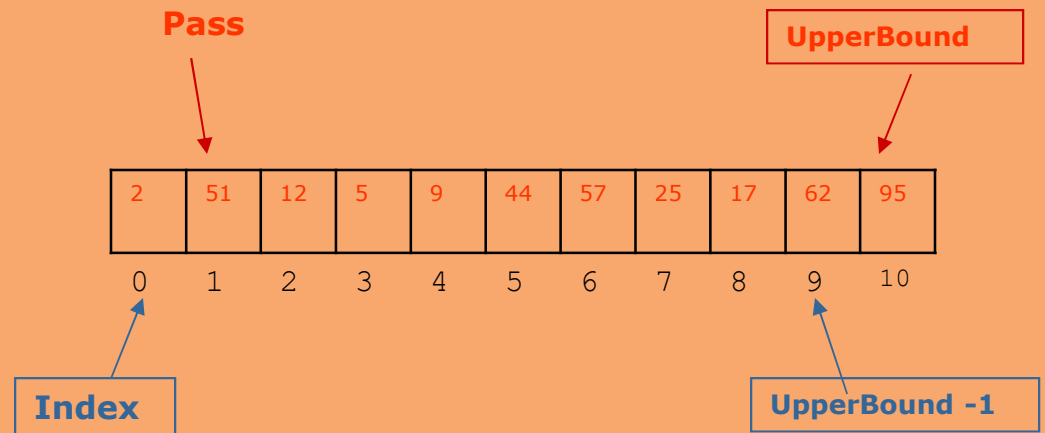
```
                TheArray(Index + 1) = Hold
```

```
            End If
```

```
        Next Index
```

```
    Next Pass
```

```
End Sub
```



Bubble Sort:

```
Private Sub BubbleSort(ByRef TheArray() As Integer)
```

```
    Dim Pass, Index, Hold As Integer
```

Hold

```
    For Pass = 1 To TheArray.GetUpperBound(0)
```

```
        For Index = 0 To TheArray.GetUpperBound(0) - 1
```

```
            If TheArray(Index) > TheArray(Index + 1) Then
```

```
                Hold = TheArray(Index)
```

```
                TheArray(Index) = TheArray(Index + 1)
```

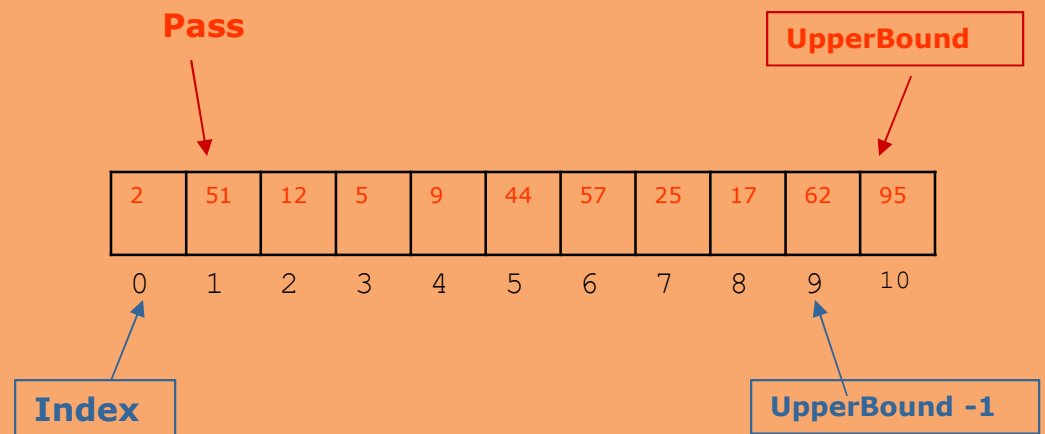
```
                TheArray(Index + 1) = Hold
```

```
            End If
```

```
        Next Index
```

```
    Next Pass
```

```
End Sub
```



Bubble Sort:

```
Private Sub BubbleSort(ByRef TheArray() As Integer)
```

```
    Dim Pass, Index, Hold As Integer
```

Hold

```
    For Pass = 1 To TheArray.GetUpperBound(0)
```

```
        For Index = 0 To TheArray.GetUpperBound(0) - 1
```

```
            If TheArray(Index) > TheArray(Index + 1) Then
```

```
                Hold = TheArray(Index)
```

```
                TheArray(Index) = TheArray(Index + 1)
```

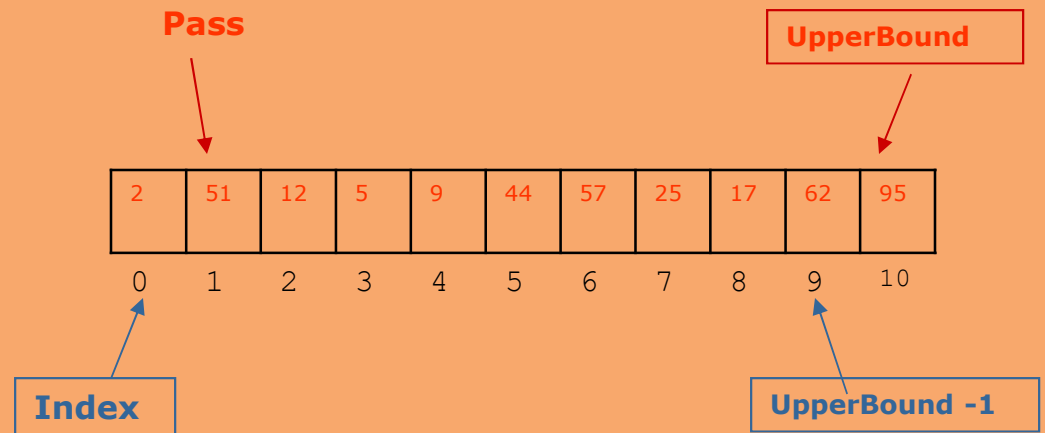
```
                TheArray(Index + 1) = Hold
```

```
            End If
```

```
        Next Index
```

```
    Next Pass
```

```
End Sub
```



Bubble Sort:

```
Private Sub BubbleSort(ByRef TheArray() As Integer)
```

```
    Dim Pass, Index, Hold As Integer
```

Hold

```
    For Pass = 1 To TheArray.GetUpperBound(0)
```

```
        For Index = 0 To TheArray.GetUpperBound(0) - 1
```

```
            If TheArray(Index) > TheArray(Index + 1) Then
```

```
                Hold = TheArray(Index)
```

```
                TheArray(Index) = TheArray(Index + 1)
```

```
                TheArray(Index + 1) = Hold
```

```
            End If
```

```
        Next Index
```

```
    Next Pass
```

```
End Sub
```

No action

Pass

UpperBound

2	51	12	5	9	44	57	25	17	62	95
0	1	2	3	4	5	6	7	8	9	10

Index

UpperBound - 1

Bubble Sort:

```
Private Sub BubbleSort(ByRef TheArray() As Integer)
```

```
    Dim Pass, Index, Hold As Integer
```

Hold

```
    For Pass = 1 To TheArray.GetUpperBound(0)
```

```
        For Index = 0 To TheArray.GetUpperBound(0) - 1
```

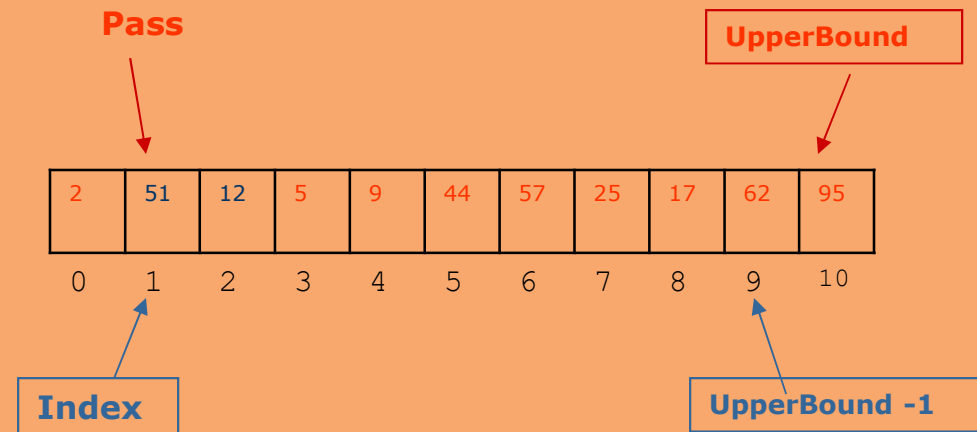
```
            If TheArray(Index) > TheArray(Index + 1) Then
                Hold = TheArray(Index)
                TheArray(Index) = TheArray(Index + 1)
                TheArray(Index + 1) = Hold
            End If
```

Swap

```
        Next Index
```

```
    Next Pass
```

```
End Sub
```



Bubble Sort:

```
Private Sub BubbleSort(ByRef TheArray() As Integer)
```

```
    Dim Pass, Index, Hold As Integer
```

Hold

51

```
    For Pass = 1 To TheArray.GetUpperBound(0)
```

```
        For Index = 0 To TheArray.GetUpperBound(0) - 1
```

```
            If TheArray(Index) > TheArray(Index + 1) Then
```

```
                Hold = TheArray(Index)
```

```
                TheArray(Index) = TheArray(Index + 1)
```

```
                TheArray(Index + 1) = Hold
```

```
            End If
```

```
        Next Index
```

```
    Next Pass
```

```
End Sub
```

Swap

Pass

UpperBound

2	12	51	5	9	44	57	25	17	62	95
0	1	2	3	4	5	6	7	8	9	10

Index

UpperBound - 1

Bubble Sort:

```
Private Sub BubbleSort(ByRef TheArray() As Integer)
```

```
    Dim Pass, Index, Hold As Integer
```

Hold

51

```
    For Pass = 1 To TheArray.GetUpperBound(0)
```

```
        For Index = 0 To TheArray.GetUpperBound(0) - 1
```

```
            If TheArray(Index) > TheArray(Index + 1) Then
```

```
                Hold = TheArray(Index)
```

```
                TheArray(Index) = TheArray(Index + 1)
```

```
                TheArray(Index + 1) = Hold
```

```
            End If
```

```
        Next Index
```

```
    Next Pass
```

```
End Sub
```

Swap

Pass

UpperBound

2	12	51	5	9	44	57	25	17	62	95
0	1	2	3	4	5	6	7	8	9	10

Index

UpperBound - 1

Bubble Sort:

```
Private Sub BubbleSort(ByRef TheArray() As Integer)
```

```
    Dim Pass, Index, Hold As Integer
```

Hold

51

```
    For Pass = 1 To TheArray.GetUpperBound(0)
```

```
        For Index = 0 To TheArray.GetUpperBound(0) - 1
```

```
            If TheArray(Index) > TheArray(Index + 1) Then
```

```
                Hold = TheArray(Index)
```

```
                TheArray(Index) = TheArray(Index + 1)
```

```
                TheArray(Index + 1) = Hold
```

```
            End If
```

```
        Next Index
```

```
    Next Pass
```

```
End Sub
```

Swap

Pass

UpperBound

2	12	5	51	9	44	57	25	17	62	95
0	1	2	3	4	5	6	7	8	9	10

Index

UpperBound - 1

Bubble Sort:

```
Private Sub BubbleSort(ByRef TheArray() As Integer)
```

```
    Dim Pass, Index, Hold As Integer
```

Hold

51

```
    For Pass = 1 To TheArray.GetUpperBound(0)
```

```
        For Index = 0 To TheArray.GetUpperBound(0) - 1
```

```
            If TheArray(Index) > TheArray(Index + 1) Then
```

```
                Hold = TheArray(Index)
```

```
                TheArray(Index) = TheArray(Index + 1)
```

```
                TheArray(Index + 1) = Hold
```

```
            End If
```

```
        Next Index
```

```
    Next Pass
```

```
End Sub
```

Swap

Pass

UpperBound

2	12	5	51	9	44	57	25	17	62	95
0	1	2	3	4	5	6	7	8	9	10

Index

UpperBound - 1

Bubble Sort:

```
Private Sub BubbleSort(ByRef TheArray() As Integer)
```

```
    Dim Pass, Index, Hold As Integer
```

Hold

51

```
    For Pass = 1 To TheArray.GetUpperBound(0)
```

```
        For Index = 0 To TheArray.GetUpperBound(0) - 1
```

```
            If TheArray(Index) > TheArray(Index + 1) Then
```

```
                Hold = TheArray(Index)
```

```
                TheArray(Index) = TheArray(Index + 1)
```

```
                TheArray(Index + 1) = Hold
```

```
            End If
```

```
        Next Index
```

```
    Next Pass
```

```
End Sub
```

Swap

Pass

UpperBound

2	12	5	9	51	44	57	25	17	62	95
0	1	2	3	4	5	6	7	8	9	10

Index

UpperBound - 1

Bubble Sort:

```
Private Sub BubbleSort(ByRef TheArray() As Integer)
```

```
    Dim Pass, Index, Hold As Integer
```

Hold

51

```
    For Pass = 1 To TheArray.GetUpperBound(0)
```

```
        For Index = 0 To TheArray.GetUpperBound(0) - 1
```

```
            If TheArray(Index) > TheArray(Index + 1) Then
```

```
                Hold = TheArray(Index)
```

```
                TheArray(Index) = TheArray(Index + 1)
```

```
                TheArray(Index + 1) = Hold
```

```
            End If
```

```
        Next Index
```

```
    Next Pass
```

```
End Sub
```

Swap

Pass

UpperBound

2	12	5	9	51	44	57	25	17	62	95
0	1	2	3	4	5	6	7	8	9	10

Index

UpperBound - 1

Bubble Sort:

```
Private Sub BubbleSort(ByRef TheArray() As Integer)
```

```
    Dim Pass, Index, Hold As Integer
```

Hold

51

```
    For Pass = 1 To TheArray.GetUpperBound(0)
```

```
        For Index = 0 To TheArray.GetUpperBound(0) - 1
```

```
            If TheArray(Index) > TheArray(Index + 1) Then
```

```
                Hold = TheArray(Index)
```

```
                TheArray(Index) = TheArray(Index + 1)
```

```
                TheArray(Index + 1) = Hold
```

```
            End If
```

```
        Next Index
```

```
    Next Pass
```

```
End Sub
```

Swap

Pass

UpperBound

2	12	5	9	44	51	57	25	17	62	95
0	1	2	3	4	5	6	7	8	9	10

Index

UpperBound - 1

Bubble Sort:

```
Private Sub BubbleSort(ByRef TheArray() As Integer)
```

```
    Dim Pass, Index, Hold As Integer
```

Hold

51

```
    For Pass = 1 To TheArray.GetUpperBound(0)
```

```
        For Index = 0 To TheArray.GetUpperBound(0) - 1
```

```
            If TheArray(Index) > TheArray(Index + 1) Then
```

```
                Hold = TheArray(Index)
```

```
                TheArray(Index) = TheArray(Index + 1)
```

```
                TheArray(Index + 1) = Hold
```

```
            End If
```

```
        Next Index
```

```
    Next Pass
```

```
End Sub
```

No Action

Pass

UpperBound

2	12	5	9	44	51	57	25	17	62	95
0	1	2	3	4	5	6	7	8	9	10

Index

UpperBound - 1

Bubble Sort:

```
Private Sub BubbleSort(ByRef TheArray() As Integer)
```

```
    Dim Pass, Index, Hold As Integer
```

Hold

57

```
    For Pass = 1 To TheArray.GetUpperBound(0)
```

```
        For Index = 0 To TheArray.GetUpperBound(0) - 1
```

```
            If TheArray(Index) > TheArray(Index + 1) Then
```

```
                Hold = TheArray(Index)
```

```
                TheArray(Index) = TheArray(Index + 1)
```

```
                TheArray(Index + 1) = Hold
```

```
            End If
```

```
        Next Index
```

```
    Next Pass
```

```
End Sub
```

Swap

Pass

UpperBound

2	12	5	9	44	51	57	25	17	62	95
0	1	2	3	4	5	6	7	8	9	10

Index

UpperBound - 1

Bubble Sort:

```
Private Sub BubbleSort(ByRef TheArray() As Integer)
```

```
    Dim Pass, Index, Hold As Integer
```

Hold

57

```
    For Pass = 1 To TheArray.GetUpperBound(0)
```

```
        For Index = 0 To TheArray.GetUpperBound(0) - 1
```

```
            If TheArray(Index) > TheArray(Index + 1) Then
```

```
                Hold = TheArray(Index)
```

```
                TheArray(Index) = TheArray(Index + 1)
```

```
                TheArray(Index + 1) = Hold
```

```
            End If
```

```
        Next Index
```

```
    Next Pass
```

```
End Sub
```

Swap

Pass

UpperBound

2	12	5	9	44	51	25	57	17	62	95
0	1	2	3	4	5	6	7	8	9	10

Index

UpperBound - 1

Bubble Sort:

```
Private Sub BubbleSort(ByRef TheArray() As Integer)
```

```
    Dim Pass, Index, Hold As Integer
```

Hold

57

```
    For Pass = 1 To TheArray.GetUpperBound(0)
```

```
        For Index = 0 To TheArray.GetUpperBound(0) - 1
```

```
            If TheArray(Index) > TheArray(Index + 1) Then
```

```
                Hold = TheArray(Index)
```

```
                TheArray(Index) = TheArray(Index + 1)
```

```
                TheArray(Index + 1) = Hold
```

```
            End If
```

```
        Next Index
```

```
    Next Pass
```

```
End Sub
```

Swap

Pass

UpperBound

2	12	5	9	44	51	25	57	17	62	95
0	1	2	3	4	5	6	7	8	9	10

Index

UpperBound - 1

Bubble Sort:

```
Private Sub BubbleSort(ByRef TheArray() As Integer)
```

```
    Dim Pass, Index, Hold As Integer
```

Hold

57

```
    For Pass = 1 To TheArray.GetUpperBound(0)
```

```
        For Index = 0 To TheArray.GetUpperBound(0) - 1
```

```
            If TheArray(Index) > TheArray(Index + 1) Then
```

```
                Hold = TheArray(Index)
```

```
                TheArray(Index) = TheArray(Index + 1)
```

```
                TheArray(Index + 1) = Hold
```

```
            End If
```

```
        Next Index
```

```
    Next Pass
```

```
End Sub
```

Swap

Pass

UpperBound

2	12	5	9	44	51	25	17	57	62	95
0	1	2	3	4	5	6	7	8	9	10

Index

UpperBound - 1

Bubble Sort:

```
Private Sub BubbleSort(ByRef TheArray() As Integer)
```

```
    Dim Pass, Index, Hold As Integer
```

Hold

57

```
    For Pass = 1 To TheArray.GetUpperBound(0)
```

```
        For Index = 0 To TheArray.GetUpperBound(0) - 1
```

```
            If TheArray(Index) > TheArray(Index + 1) Then
```

```
                Hold = TheArray(Index)
```

```
                TheArray(Index) = TheArray(Index + 1)
```

```
                TheArray(Index + 1) = Hold
```

```
            End If
```

```
        Next Index
```

```
    Next Pass
```

```
End Sub
```

No action

Pass

UpperBound

2	12	5	9	44	51	25	17	57	62	95
0	1	2	3	4	5	6	7	8	9	10

Index

UpperBound - 1

Bubble Sort:

```
Private Sub BubbleSort(ByRef TheArray() As Integer)
```

```
    Dim Pass, Index, Hold As Integer
```

Hold

57

```
    For Pass = 1 To TheArray.GetUpperBound(0)
```

```
        For Index = 0 To TheArray.GetUpperBound(0) - 1
```

```
            If TheArray(Index) > TheArray(Index + 1) Then
```

```
                Hold = TheArray(Index)
```

```
                TheArray(Index) = TheArray(Index + 1)
```

```
                TheArray(Index + 1) = Hold
```

```
            End If
```

```
        Next Index
```

```
    Next Pass
```

```
End Sub
```

No action

Pass

UpperBound

2	12	5	9	44	51	25	17	57	62	95
0	1	2	3	4	5	6	7	8	9	10

Index

UpperBound - 1

End of First Pass

□ Pass = 2

Bubble Sort:

```
Private Sub BubbleSort(ByRef TheArray() As Integer)
```

```
    Dim Pass, Index, Hold As Integer
```

Hold

57

```
    For Pass = 1 To TheArray.GetUpperBound(0)
```

```
        For Index = 0 To TheArray.GetUpperBound(0) - 1
```

```
            If TheArray(Index) > TheArray(Index + 1) Then
```

```
                Hold = TheArray(Index)
```

```
                TheArray(Index) = TheArray(Index + 1)
```

```
                TheArray(Index + 1) = Hold
```

```
            End If
```

```
        Next Index
```

```
    Next Pass
```

```
End Sub
```

No action

Pass

UpperBound

2	12	5	9	44	51	25	17	57	62	95
0	1	2	3	4	5	6	7	8	9	10

Index

UpperBound - 1

Bubble Sort:

```
Private Sub BubbleSort(ByRef TheArray() As Integer)
```

```
    Dim Pass, Index, Hold As Integer
```

Hold

12

```
    For Pass = 1 To TheArray.GetUpperBound(0)
```

```
        For Index = 0 To TheArray.GetUpperBound(0) - 1
```

```
            If TheArray(Index) > TheArray(Index + 1) Then
```

```
                Hold = TheArray(Index)
```

```
                TheArray(Index) = TheArray(Index + 1)
```

```
                TheArray(Index + 1) = Hold
```

```
            End If
```

```
        Next Index
```

```
    Next Pass
```

```
End Sub
```

swap

Pass

UpperBound

2	12	5	9	44	51	25	17	57	62	95
0	1	2	3	4	5	6	7	8	9	10

Index

UpperBound - 1

Bubble Sort:

```
Private Sub BubbleSort(ByRef TheArray() As Integer)
```

```
    Dim Pass, Index, Hold As Integer
```

Hold

12

```
    For Pass = 1 To TheArray.GetUpperBound(0)
```

```
        For Index = 0 To TheArray.GetUpperBound(0) - 1
```

```
            If TheArray(Index) > TheArray(Index + 1) Then
```

```
                Hold = TheArray(Index)
```

```
                TheArray(Index) = TheArray(Index + 1)
```

```
                TheArray(Index + 1) = Hold
```

```
            End If
```

```
        Next Index
```

```
    Next Pass
```

```
End Sub
```

swap

Pass

UpperBound

2	5	12	9	44	51	25	17	57	62	95
0	1	2	3	4	5	6	7	8	9	10

Index

UpperBound - 1

Bubble Sort:

```
Private Sub BubbleSort(ByRef TheArray() As Integer)
```

```
    Dim Pass, Index, Hold As Integer
```

Hold

12

```
    For Pass = 1 To TheArray.GetUpperBound(0)
```

```
        For Index = 0 To TheArray.GetUpperBound(0) - 1
```

```
            If TheArray(Index) > TheArray(Index + 1) Then
```

```
                Hold = TheArray(Index)
```

```
                TheArray(Index) = TheArray(Index + 1)
```

```
                TheArray(Index + 1) = Hold
```

```
            End If
```

```
        Next Index
```

```
    Next Pass
```

```
End Sub
```

swap

Pass

UpperBound

2	5	12	9	44	51	25	17	57	62	95
0	1	2	3	4	5	6	7	8	9	10

Index

UpperBound - 1

Bubble Sort:

```
Private Sub BubbleSort(ByRef TheArray() As Integer)
```

```
    Dim Pass, Index, Hold As Integer
```

Hold

12

```
    For Pass = 1 To TheArray.GetUpperBound(0)
```

```
        For Index = 0 To TheArray.GetUpperBound(0) - 1
```

```
            If TheArray(Index) > TheArray(Index + 1) Then
```

```
                Hold = TheArray(Index)
```

```
                TheArray(Index) = TheArray(Index + 1)
```

```
                TheArray(Index + 1) = Hold
```

```
            End If
```

```
        Next Index
```

```
    Next Pass
```

```
End Sub
```

swap

Pass

UpperBound

2	5	9	12	44	51	25	17	57	62	95
0	1	2	3	4	5	6	7	8	9	10

Index

UpperBound - 1

Bubble Sort:

```
Private Sub BubbleSort(ByRef TheArray() As Integer)
```

```
    Dim Pass, Index, Hold As Integer
```

Hold

12

```
    For Pass = 1 To TheArray.GetUpperBound(0)
```

```
        For Index = 0 To TheArray.GetUpperBound(0) - 1
```

```
            If TheArray(Index) > TheArray(Index + 1) Then
```

```
                Hold = TheArray(Index)
```

```
                TheArray(Index) = TheArray(Index + 1)
```

```
                TheArray(Index + 1) = Hold
```

```
            End If
```

```
        Next Index
```

```
    Next Pass
```

```
End Sub
```

No action

Pass

UpperBound

2	5	9	12	44	51	25	17	57	62	95
0	1	2	3	4	5	6	7	8	9	10

Index

UpperBound - 1

Bubble Sort:

```
Private Sub BubbleSort(ByRef TheArray() As Integer)
```

```
    Dim Pass, Index, Hold As Integer
```

Hold

12

```
    For Pass = 1 To TheArray.GetUpperBound(0)
```

```
        For Index = 0 To TheArray.GetUpperBound(0) - 1
```

```
            If TheArray(Index) > TheArray(Index + 1) Then
```

```
                Hold = TheArray(Index)
```

```
                TheArray(Index) = TheArray(Index + 1)
```

```
                TheArray(Index + 1) = Hold
```

```
            End If
```

```
        Next Index
```

```
    Next Pass
```

```
End Sub
```

No action

Pass

UpperBound

2	5	9	12	44	51	25	17	57	62	95
0	1	2	3	4	5	6	7	8	9	10

Index

UpperBound - 1

□ And so on...

MIN and MAX

```
//-----  
public int min() { // Calculate the min  
  
    int the_min = theArray[0]; // assume the first element is the smallest  
  
    for (int i = 1; i < nElems; i++) {  
        if (the_min > theArray[i]) {  
            the_min = theArray[i];  
        }  
    }  
    return (the_min);  
}
```

```
//-----  
public int max() { // Calculate the max  
    int the_max = theArray[0]; // assume the first element is the largest  
    for (int i = 1; i < nElems; i++) {  
        if (the_max < theArray[i]) {  
            the_max = theArray[i];  
        }  
    }  
    return (the_max);  
}
```

MEAN

```
//-----  
// Mean()  
public double mean() {    // calculate the average  
    double sum = 0;  
    for (int i = 0; i < nElems; i++) {  
        sum = sum + theArray[i];  
    }  
    return (sum / nElems);  
}
```

StdDev

$$s = \sqrt{\frac{\sum_{i=1}^N (x_i - \bar{x})^2}{N - 1}}$$

https://en.wikipedia.org/wiki/Standard_deviation

- is a measure that is used to quantify the amount of variation or dispersion of a set of data values.
- A low standard deviation indicates that the data points tend to be close to the mean.
- a high standard deviation indicates that the data points are spread out over a wider range of values.

Mean and StdDev

```
//-----  
// Mean()  
public double mean() {    // calculate the average  
    double sum = 0;  
    for (int i = 0; i < nElems; i++) {  
        sum = sum + theArray[i];  
    }  
    return (sum / nElems);  
}
```

```
//-----  
// StdDev()  
public double stdDev() {    // Calculate the Standard Deviation  
  
    double term = 0.0;  
    double summation = 0.0;  
    double array_element;  
    double the_mean, the_std_dev;  
  
    the_mean = mean();  
  
    //System.out.println("Mean = " + the_mean);  
  
    for (int i = 0; i < nElems; i++) {  
        array_element = (double) theArray[i];  
        term = Math.pow(array_element - the_mean, 2.0);  
        summation = summation + term;  
    }  
  
    the_std_dev = Math.sqrt(summation / (nElems - 1));  
  
    return (the_std_dev);  
}
```

Mode

- ❑ The mode of a set of data values is the value that appears most often.
- ❑ The mode is not necessarily unique to a given discrete distribution, since the probability mass function may take the same maximum value at several points x_1, x_2 , etc. The most extreme case occurs in uniform distributions, where all values occur equally frequently.

	Values	Sorted Values
	77	-1
	99	0
	44	11
	55	11
	-1	33
	88	44
	11	55
	0	55
	66	66
	33	77
	11	88
	55	99
Mean	44.83333333	44.83333333
Median	49.5	49.5
Mode	55	11
Stddev	34.41414438	34.41414438
Min	-1	-1
Max	99	99

MODE

```
public int mode() { // Calculate the mode
    int currentValue, currentCount;
    int maxValue = 0;
    int maxCount = 0;

    int[] counts = new int[nElems];

    for (int i = 0; i < nElems; i++) {
        currentValue = theArray[i]; // get the current value
        currentCount = freqCount(currentValue); // see how many there are array

        if (currentCount > maxCount) {
            System.out.println("Candidate Mode: " + currentValue + ", Count= " + currentCount);
            maxCount = currentCount;
            maxValue = currentValue;
        }
    }
    return maxValue;
}
//-----
public int freqCount(int theValue) { // Calculate the freqCount for a given value

    int count = 0;
    for (int i = 0; i < nElems; i++) {
        if(theArray[i] == theValue)
            count++;
    }
    return count;
}
```

Median

- The median is the value separating the higher half from the lower half of a data sample

1, 3, 3, **6**, 7, 8, 9

Median = **6**

1, 2, 3, **4**, **5**, 6, 8, 9

Median = $(4 + 5) \div 2$
= **4.5**

<https://en.wikipedia.org/wiki/Median>