

Week 5 Homework Q2

Telmen Enkhbold

San Fransico Bay University

CE480 - Java and Internet Application

Dr. Chang, Henry

10/12/2023

Author Note

The Question

Use array element as an index to another array element.

- Step 1: Design a program to compute the histogram of a 2D array and display the result. Where the histogram is the frequency counts of the population of elements of each array value. For example, the histogram of the following 10 * 10 array. (Assume the range of the array value is from 0 to 255)

```

○
○      1  3  5  7  9  3  4  4  5  6
○      1  3  5  7  9  3  4  4  5  6
○      1  3  5  7  9  3  4  4  5  6
○      1  3  5 20 25 24 33  5  6  4
○      1  3  5 22 35 24 32  5  6  4
○      1  3  5 20 28 34 23  5  6  4
○      1  3  5 21 25 27 23  5  6  4
○      1  3  5  7  9  3  4  4  5  6
○      1  3  5  7  9  3  4  4  5  6
○      1  3  5  7  9  3  4  4  5  6
○

```

is

```

○
○      gray-level      #-of-pixels
○      1              10
○      3              16
○      4              16
○      5              20
○      6              10
○      7              6
○      9              6
○      20             2
○      21             1
○      22             1
○      23             2
○      24             2
○      25             2
○      27             1
○      28             1
○      32             1
○      33             1
○      34             1
○      35             1

```

Note:

- Sample C code

You only need to have two arrays in your program

```

#define max 256
#define N 10
void main()
{
    /* Image data */
    int image[N][N]={
        {1,3,5,7,9,3,4,4,5,6},
        {1,3,5,7,9,3,4,4,5,6},
        {1,3,5,7,9,3,4,4,5,6},
        {1,3,5,20,25,24,33,5,6,4},
        {1,3,5,20,35,24,32,5,6,4},
        {1,3,5,20,28,34,23,5,6,4},
        {1,3,5,21,25,27,23,5,6,4},
        {1,3,5,7,9,3,4,4,5,6},
        {1,3,5,7,9,3,4,4,5,6},
        {1,3,5,7,9,3,4,4,5,6}
    };
    /* Histogram of the image data */
    int hist[max]={0};

    /* The code size of calculating a histogram from
     * an image data is less than 5 lines
     */
    ....
    ....
}

```

- Step 2: Please also verify your program with this image

```

○
○ 1  3  5  7  9  3  4  4  5  6
○ 1 20 25 24 3  5  6  4  2  4
○ 1 22 35 24 3  5  6  4  5  7
○ 1 20 28 34 2  5  6  4  8  9
○ 1  3  5  7  9  3  4  4  5  6
○ 1  3  5  7  9  3 67  4  5  6
○ 1  3  5  7  9 78 54 94  5  6
○ 1  3  5  7  9 99 98 54  5  6
○ 1  3  5  7  9  3 64  4  5  6
○ 1  3  5  7  9  3  4  4  5  6

```

- Step 3: Implement the code using object-oriented approach. For example, for Java

```

○
○ class Image
○ {
○     private int[][] pixels;
○
○     public Image(int[][] pixels) {
○         ....
○     }
○
○     public int[] histogram() {

```

```

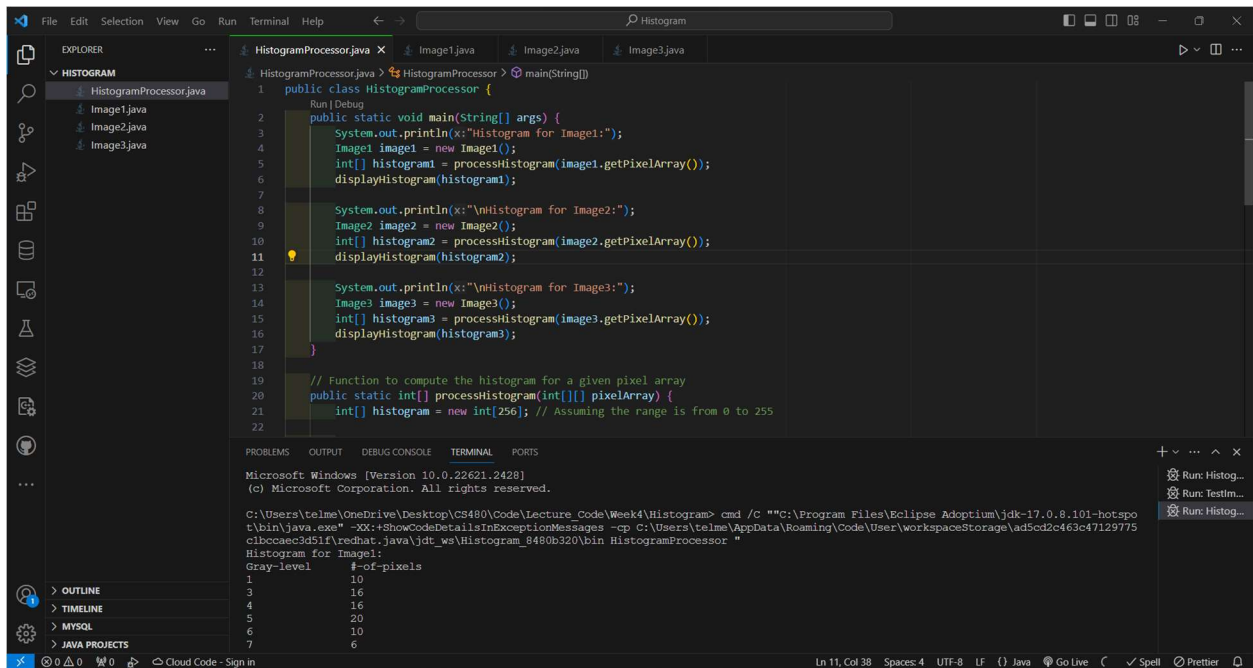
o         ....
o
o         return histogram;
o     }
o }
o }
o public class TestImage {
o     public static void main(String[] args) {
o         int[][] pixelArray = {
o             {1, 3, 5, 7, 9, 3, 4, 4, 5, 6},
o             {1, 3, 5, 7, 9, 3, 4, 4, 5, 6},
o             {1, 3, 5, 7, 9, 3, 4, 4, 5, 6},
o             {1, 3, 5, 20, 25, 24, 33, 5, 6, 4},
o             {1, 3, 5, 22, 35, 24, 32, 5, 6, 4},
o             {1, 3, 5, 20, 28, 34, 23, 5, 6, 4},
o             {1, 3, 5, 21, 25, 27, 23, 5, 6, 4},
o             {1, 3, 5, 7, 9, 3, 4, 4, 5, 6},
o             {1, 3, 5, 7, 9, 3, 4, 4, 5, 6},
o             {1, 3, 5, 7, 9, 3, 4, 4, 5, 6}
o         };
o
o         Image image = new Image(pixelArray);
o         .....
o     }

```

Note:

- o You can get a [hint](#) from ChatGPT
- o References
 - [Yuhong Luo](#) - Java implementation, 2019 Fall
 - [The histogram of image intentisty](#) ([local copy](#))
- o [MapReduce solution](#)

Screenshot



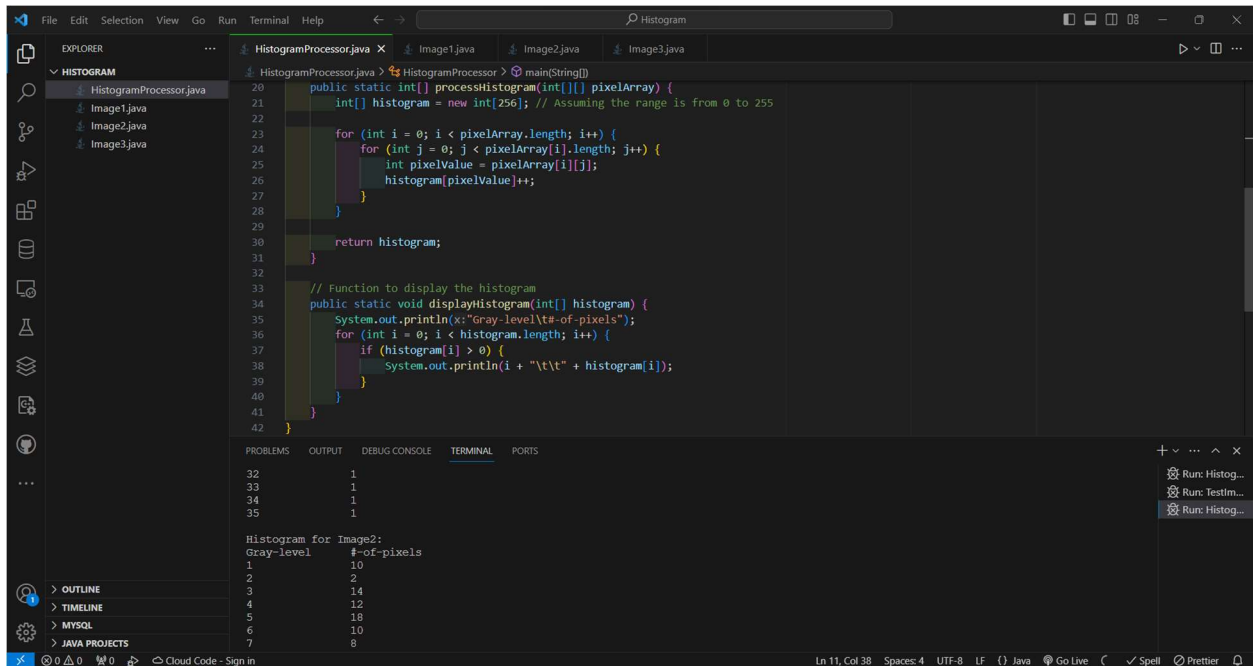
```
1 public class HistogramProcessor {
2     public static void main(String[] args) {
3         System.out.println("Histogram for Image1:");
4         Image1 image1 = new Image1();
5         int[] histogram1 = processHistogram(image1.getPixelArray());
6         displayHistogram(histogram1);
7
8         System.out.println("\nHistogram for Image2:");
9         Image2 image2 = new Image2();
10        int[] histogram2 = processHistogram(image2.getPixelArray());
11        displayHistogram(histogram2);
12
13        System.out.println("\nHistogram for Image3:");
14        Image3 image3 = new Image3();
15        int[] histogram3 = processHistogram(image3.getPixelArray());
16        displayHistogram(histogram3);
17    }
18
19    // Function to compute the histogram for a given pixel array
20    public static int[] processHistogram(int[][] pixelArray) {
21        int[] histogram = new int[256]; // Assuming the range is from 0 to 255
22    }
```

Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

C:\Users\telme\OneDrive\Desktop\CS480\Code\Lecture_Code\Week4\Histogram> cmd /C ""C:\Program Files\Eclipse Adoptium\jdk-17.0.8-hotspot\bin\java.exe" -XX:+ShowCodeDetailsInExceptionMessages -cp C:\Users\telme\AppData\Roaming\Code\User\workspaceStorage\ad5cd2c463c47129775\cibc0a0c3d51f\redhat_java\jdt_ws\Histogram_8480b320\bin HistogramProcessor "

Histogram for Image1:
Gray-level #of-pixels

Gray-level	#of-pixels
1	10
3	16
4	16
5	20
6	10
7	6



```
20 public static int[] processHistogram(int[][] pixelArray) {
21     int[] histogram = new int[256]; // Assuming the range is from 0 to 255
22
23     for (int i = 0; i < pixelArray.length; i++) {
24         for (int j = 0; j < pixelArray[i].length; j++) {
25             int pixelValue = pixelArray[i][j];
26             histogram[pixelValue]++;
27         }
28     }
29
30     return histogram;
31 }
32
33 // Function to display the histogram
34 public static void displayHistogram(int[] histogram) {
35     System.out.println("Gray-level\t#-of-pixels");
36     for (int i = 0; i < histogram.length; i++) {
37         if (histogram[i] > 0) {
38             System.out.println(i + "\t\t" + histogram[i]);
39         }
40     }
41 }
42 }
```

Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

C:\Users\telme\OneDrive\Desktop\CS480\Code\Lecture_Code\Week4\Histogram> cmd /C ""C:\Program Files\Eclipse Adoptium\jdk-17.0.8-hotspot\bin\java.exe" -XX:+ShowCodeDetailsInExceptionMessages -cp C:\Users\telme\AppData\Roaming\Code\User\workspaceStorage\ad5cd2c463c47129775\cibc0a0c3d51f\redhat_java\jdt_ws\Histogram_8480b320\bin HistogramProcessor "

Histogram for Image2:
Gray-level #of-pixels

Gray-level	#of-pixels
1	10
2	2
3	14
4	12
5	18
6	10
7	8

```
Image1.java > ...
1 public class Image1 {
2     private int[][] pixelArray = {
3         {1, 3, 5, 7, 9, 3, 4, 4, 5, 6},
4         {1, 3, 5, 7, 9, 3, 4, 4, 5, 6},
5         {1, 3, 5, 7, 9, 3, 4, 4, 5, 6},
6         {1, 3, 5, 20, 25, 24, 33, 5, 6, 4},
7         {1, 3, 5, 22, 35, 24, 32, 5, 6, 4},
8         {1, 3, 5, 20, 28, 34, 23, 5, 6, 4},
9         {1, 3, 5, 21, 25, 27, 23, 5, 6, 4},
10        {1, 3, 5, 7, 9, 3, 4, 4, 5, 6},
11        {1, 3, 5, 7, 9, 3, 4, 4, 5, 6},
12        {1, 3, 5, 7, 9, 3, 4, 4, 5, 6}
13    };
14
15    public int[][] getPixelArray() {
16        return pixelArray;
17    }
18 }
19
```

```
Image2.java > ...
1 public class Image2 {
2     private int[][] pixelArray = {
3         {1, 3, 5, 7, 9, 3, 4, 4, 5, 6},
4         {1, 20, 25, 24, 3, 5, 6, 4, 2, 4},
5         {1, 22, 35, 24, 3, 5, 6, 4, 5, 7},
6         {1, 20, 28, 34, 2, 5, 6, 4, 8, 9},
7         {1, 3, 5, 7, 9, 3, 4, 4, 5, 6},
8         {1, 3, 5, 7, 9, 3, 67, 4, 5, 6},
9         {1, 3, 5, 7, 9, 78, 54, 94, 5, 6},
10        {1, 3, 5, 7, 9, 99, 98, 54, 5, 6},
11        {1, 3, 5, 7, 9, 3, 64, 4, 5, 6},
12        {1, 3, 5, 7, 9, 3, 4, 4, 5, 6}
13    };
14
15    public int[][] getPixelArray() {
16        return pixelArray;
17    }
18 }
19
```

```

Image3.java > ...
1  public class Image3 {
2      private int[][] pixelArray = {
3          {1, 3, 5, 7, 9, 3, 4, 4, 5, 6},
4          {1, 3, 5, 7, 9, 3, 4, 4, 5, 6},
5          {1, 3, 5, 7, 9, 3, 4, 4, 5, 6},
6          {1, 3, 5, 20, 25, 24, 33, 5, 6, 4},
7          {1, 3, 5, 22, 35, 24, 32, 5, 6, 4},
8          {1, 3, 5, 20, 28, 34, 23, 5, 6, 4},
9          {1, 3, 5, 21, 25, 27, 23, 5, 6, 4},
10         {1, 3, 5, 7, 9, 3, 4, 4, 5, 6},
11         {1, 3, 5, 7, 9, 3, 4, 4, 5, 6},
12         {1, 3, 5, 7, 9, 3, 4, 4, 5, 6}
13     };
14
15     public int[][] getPixelArray() {
16         return pixelArray;
17     }
18 }
19

```

The Source Code

-----Histogram Processor-----

```

public class HistogramProcessor {
    public static void main(String[] args) {
        System.out.println("Histogram for Image1:");
        Image1 image1 = new Image1();
        int[] histogram1 = processHistogram(image1.getPixelArray());
        displayHistogram(histogram1);

        System.out.println("\nHistogram for Image2:");
        Image2 image2 = new Image2();
        int[] histogram2 = processHistogram(image2.getPixelArray());
        displayHistogram(histogram2);

        System.out.println("\nHistogram for Image3:");
        Image3 image3 = new Image3();
        int[] histogram3 = processHistogram(image3.getPixelArray());
        displayHistogram(histogram3);
    }
}

```

```

    }

    // Function to compute the histogram for a given pixel array
    public static int[] processHistogram(int[][] pixelArray) {
        int[] histogram = new int[256]; // Assuming the range is from 0 to
255

        for (int i = 0; i < pixelArray.length; i++) {
            for (int j = 0; j < pixelArray[i].length; j++) {
                int pixelValue = pixelArray[i][j];
                histogram[pixelValue]++;
            }
        }

        return histogram;
    }

    // Function to display the histogram
    public static void displayHistogram(int[] histogram) {
        System.out.println("Gray-level\t#-of-pixels");
        for (int i = 0; i < histogram.length; i++) {
            if (histogram[i] > 0) {
                System.out.println(i + "\t\t" + histogram[i]);
            }
        }
    }
}

```

-----Image1-----

```

public class Image1 {
    private int[][] pixelArray = {
        {1, 3, 5, 7, 9, 3, 4, 4, 5, 6},
        {1, 3, 5, 7, 9, 3, 4, 4, 5, 6},
        {1, 3, 5, 7, 9, 3, 4, 4, 5, 6},
        {1, 3, 5, 20, 25, 24, 33, 5, 6, 4},
        {1, 3, 5, 22, 35, 24, 32, 5, 6, 4},
        {1, 3, 5, 20, 28, 34, 23, 5, 6, 4},
        {1, 3, 5, 21, 25, 27, 23, 5, 6, 4},
        {1, 3, 5, 7, 9, 3, 4, 4, 5, 6},
        {1, 3, 5, 7, 9, 3, 4, 4, 5, 6},
        {1, 3, 5, 7, 9, 3, 4, 4, 5, 6}
    };
}

```



```

public int[][] getPixelArray() {
    return pixelArray;
}
}

```

-----Image2-----

```

public class Image2 {
    private int[][] pixelArray = {
        {1, 3, 5, 7, 9, 3, 4, 4, 5, 6},
        {1, 20, 25, 24, 3, 5, 6, 4, 2, 4},
        {1, 22, 35, 24, 3, 5, 6, 4, 5, 7},
        {1, 20, 28, 34, 2, 5, 6, 4, 8, 9},
        {1, 3, 5, 7, 9, 3, 4, 4, 5, 6},
        {1, 3, 5, 7, 9, 3, 67, 4, 5, 6},
        {1, 3, 5, 7, 9, 78, 54, 94, 5, 6},
        {1, 3, 5, 7, 9, 99, 98, 54, 5, 6},
        {1, 3, 5, 7, 9, 3, 64, 4, 5, 6},
        {1, 3, 5, 7, 9, 3, 4, 4, 5, 6}
    };

    public int[][] getPixelArray() {
        return pixelArray;
    }
}

```

-----Image3-----

```

public class Image3 {
    private int[][] pixelArray = {
        {1, 3, 5, 7, 9, 3, 4, 4, 5, 6},
        {1, 3, 5, 7, 9, 3, 4, 4, 5, 6},
        {1, 3, 5, 7, 9, 3, 4, 4, 5, 6},
        {1, 3, 5, 20, 25, 24, 33, 5, 6, 4},
        {1, 3, 5, 22, 35, 24, 32, 5, 6, 4},
        {1, 3, 5, 20, 28, 34, 23, 5, 6, 4},
        {1, 3, 5, 21, 25, 27, 23, 5, 6, 4},
        {1, 3, 5, 7, 9, 3, 4, 4, 5, 6},
    };
}

```

```
        {1, 3, 5, 7, 9, 3, 4, 4, 5, 6},  
        {1, 3, 5, 7, 9, 3, 4, 4, 5, 6}  
    };  
  
    public int[][] getPixelArray() {  
        return pixelArray;  
    }  
}
```