Gebze Technical University
Computer Engineering

CSE 222 - 2019 Spring

HOMEWORK 5 REPORT

NEVRA GÜRSES
161044071

Özgü Göksu

1 INTRODUCTION

1.1 Problem Definition

There is a color digital image in a file.We have to read every pixel of this image and then we have to represent its all pixels in a 2D array.Every element of 2D array is a pixel and this pixels includes 3D 8 bit unsigned int valued vectors.Every 8 bit is amount of one color (first dimention is red,second dimention is green and third dimention is blue) in any pixel and range of one 8bit is 0-255. After reading color image , we have to do max priority queue for this color pixels.We have to add pixels in max priority queue one by one and extract them different priority ordering relations with 3 comparision method (lexicographical comparison, Euclidean norm based comparison and bitmix comparison).This adding and extracting operations is doing 4 threads as concurrently.First 100 pixel is added in 3 priority queue by thread 1.After that, 3 thread is created and they extracts pixels from different priority queue.(thread 2 extracts from lex ordering priority queue,thread 3 extracts from euc ordering priority queue and thread 4 extracts from bitmix ordering priority queue.)
While extracting operations is doing by 3 threads concurrently,thread 1 continues adding in every queue other pixels until all pixels added.

1.2 System Requirements

This program is writed with Java programming language.And Java requires following system requirements for Windows:
→Windows 10 (7u85 and above)
→Windows 8.x (Desktop)
→Windows 7 SP1.
→ Vista SP2.
→Windows Server 2008 SP2 and 2008 R2 SP1 (64-bit)
→Windows Server 2012 (64-bit) and 2012 R2 (64-bit)
→RAM: 128 MB; 64 MB for Windows XP (32-bit)
→Disk space: 124 MB.
This program is writed in Intellij IDEA.And Intellij IDEA requires following system requiremets:
→Windows 10/8/7/Vista/XP (incl. 64-bit).

→2 GB RAM minimum, 4 GB RAM recommended.

→1.5 GB hard disk space + at least 1 GB for caches.

→1024x768 minimum screen resolution.

This program also require Java SE Development Kit 11.0.2 .Java SE Development Kit (JDK) and Java SE Runtime Environment (JRE) require at minimum a Pentium 2 266 MHz processor.For the JDK, the option of installing the following features:

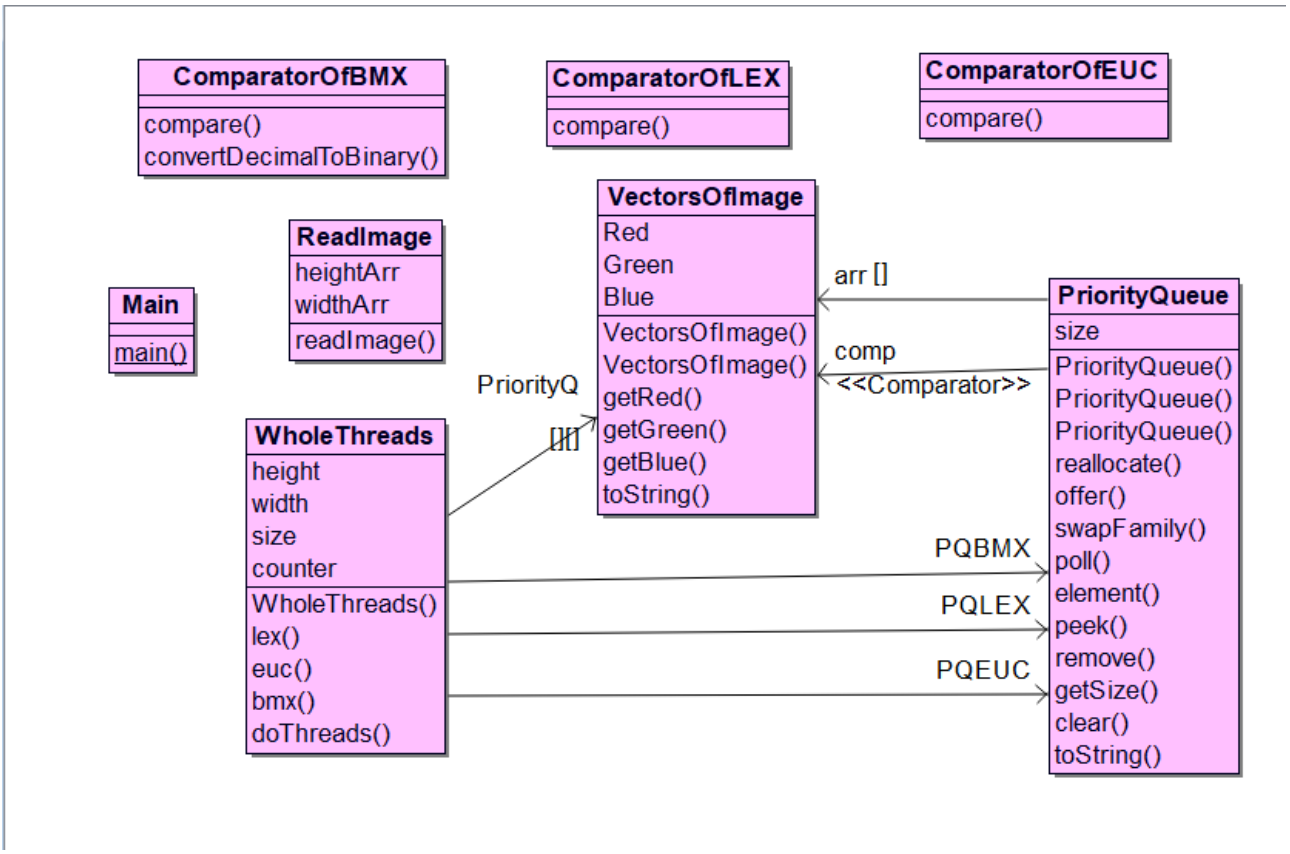
→Development Tools.

→Source Code.

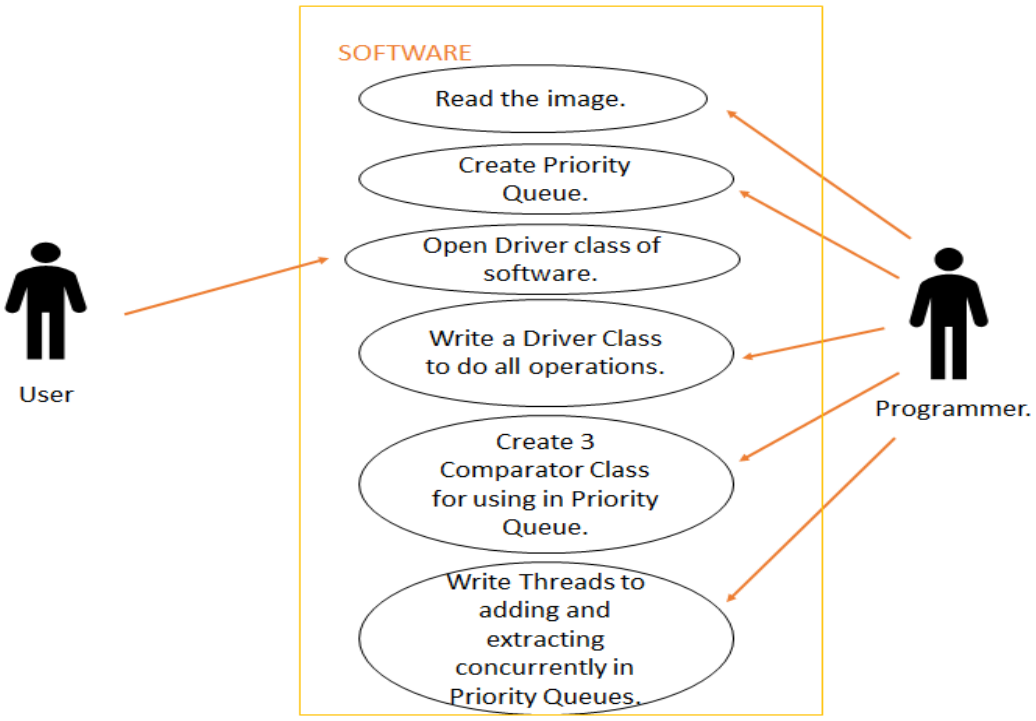
→Public Java Runtime Environment.

2 METHOD

2.1 Class Diagrams



2.2 Use Case Diagrams



2.3 Problem Solution Approach

To solve this problem , Firstly I read the image with using BufferdImage and I register every pixel in 2D array.Data type of 2D array is a class type named VectorsOfImage and that class has int data fields for amount of red color,green color and blue color of every pixel.After that I write max priority queue that adds and extracts pixels that their type is VectorsOfImage class type.To ordering pixels with different priority , I create 3 comparator class that have different comparing purpose.First comparator compares two pixels according to lexicographical magnitude,second comparator compares two pixels according to Euclidean norm based magnitude and third comparator compares two pixels according to bitmix magnitude.To use compare method of this comparators in my priority queue, I create a data field its type is Comparator and I write a constructor that takes a comparator as parameter.So I initilize my comparator data field with given comparator type.So actually,3 priority queue occurs by different compare methods.In this way,thread1 can add pixels in every priority queue and other threads can extract pixels from priority queues according to different ordering relation.

Time complexity of this program O(nlogn).Explain this:

I read the image and record all pixels of image in a 2D array.This array has n element(height*width of image).After that, in WholeThreads class,all pixels is recorded in priority queues and then extracting them.The offer method and the poll method of Priority Queue is O(logn) So time complexity of adding and extracting all pixels by Threads is O(nlogn).And also total time complexity of program is O(nlogn).

3 RESULT

3.1 Test Cases

Firstly, I controled offer method in my priority queue with giving different pixels and I tested all compare methods with printing priorty queue in screen.Then I control poll and other methods in my priorty queue with printing screen.They were working true.Then I did all tasks in homework and printed screen one by one.Threads were working true.First 100 pixel is only adding 3 priority queue and than other threads extract from pixels in queues as simultaneously and also remaining pixels were adding in queues by Thread 1.

3.2 Running Results

First 100 pixel added 3 priority queue by thread 1.

```
"C:\Program Files\Java\jdk-11.0.2\bin\
Thread 1: [ 200 , 159 , 0 ]
Thread 1: [ 200 , 159 , 0 ]
Thread 1: [ 200 , 159 , 0 ]
Thread 1: [ 200 , 159 , 0 ]
Thread 1: [ 201 , 159 , 0 ]
Thread 1: [ 201 , 159 , 0 ]
Thread 1: [ 201 , 159 , 0 ]
Thread 1: [ 202 , 158 , 0 ]
Thread 1: [ 202 , 158 , 0 ]
Thread 1: [ 202 , 158 , 0 ]
Thread 1: [ 202 , 158 , 0 ]
Thread 1: [ 202 , 158 , 0 ]
Thread 1: [ 202 , 158 , 0 ]
Thread 1: [ 202 , 158 , 0 ]
Thread 1: [ 203 , 157 , 0 ]
Thread 1: [ 203 , 157 , 0 ]
Thread 1: [ 203 , 157 , 0 ]
Thread 1: [ 204 , 157 , 0 ]
Thread 1: [ 204 , 157 , 0 ]
Thread 1: [ 204 , 157 , 0 ]
Thread 1: [ 204 , 157 , 0 ]
Thread 1: [ 204 , 157 , 0 ]
Thread 1: [ 204 , 157 , 0 ]
Thread 1: [ 204 , 157 , 0 ]
Thread 1: [ 204 , 157 , 0 ]
Thread 1: [ 204 , 156 , 0 ]
Thread 1: [ 205 , 156 , 1 ]
Thread 1: [ 205 , 156 , 1 ]
Thread 1: [ 205 , 156 , 1 ]
Thread 1: [ 206 , 156 , 2 ]
Thread 1: [ 206 , 156 , 2 ]
Thread 1: [ 206 , 155 , 2 ]
Thread 1: [ 206 , 155 , 2 ]
Thread 1: [ 206 , 155 , 2 ]
Thread 1: [ 206 , 155 , 2 ]
Thread 1: [ 207 , 155 , 3 ]
Thread 1: [ 207 , 155 , 3 ]
Thread 1: [ 207 , 155 , 3 ]
Thread 1: [ 208 , 154 , 4 ]
Thread 1: [ 208 , 154 , 4 ]
Thread 1: [ 208 , 154 , 4 ]
Thread 1: [ 208 , 154 , 4 ]
Thread 1: [ 208 , 154 , 4 ]
Thread 1: [ 208 , 154 , 4 ]
Thread 1: [ 208 , 154 , 4 ]
Thread 1: [ 208 , 154 , 4 ]
Thread 1: [ 208 , 153 , 4 ]
Thread 1: [ 209 , 153 , 5 ]
Thread 1: [ 209 , 153 , 5 ]
Thread 1: [ 209 , 153 , 5 ]
Thread 1: [ 210 , 153 , 6 ]
Thread 1: [ 210 , 153 , 6 ]
Thread 1: [ 210 , 153 , 6 ]
Thread 1: [ 210 , 152 , 6 ]
Thread 1: [ 210 , 152 , 6 ]
Thread 1: [ 210 , 152 , 6 ]
Thread 1: [ 210 , 152 , 6 ]
Thread 1: [ 211 , 152 , 7 ]
Thread 1: [ 211 , 152 , 7 ]
Thread 1: [ 211 , 152 , 7 ]
```

After 100 pixel added,3 threads created and every thread do own task.

```
Thread 1: [ 212 , 151 , 8 ]
Thread 1: [ 212 , 151 , 8 ]
Thread 1: [ 212 , 151 , 8 ]
Thread 1: [ 212 , 151 , 8 ]
Thread 1: [ 213 , 151 , 9 ]
Thread 1: [ 213 , 151 , 9 ]
Thread 1: [ 213 , 151 , 9 ]
Thread 1: [ 213 , 150 , 9 ]
Thread 1: [ 213 , 150 , 9 ]
Thread 1: [ 213 , 150 , 9 ]
Thread 1: [ 213 , 150 , 9 ]
Thread 1: [ 214 , 150 , 10 ]
Thread 1: [ 214 , 150 , 10 ]
Thread 1: [ 214 , 150 , 10 ]
Thread 1: [ 215 , 150 , 11 ]
Thread 1: [ 215 , 149 , 11 ]
Thread 1: [ 215 , 149 , 11 ]
Thread 1: [ 215 , 149 , 11 ]
Thread 1: [ 216 , 148 , 12 ]
Thread 1: [ 216 , 148 , 12 ]
Thread 1: [ 216 , 148 , 12 ]
Thread 1: [ 216 , 148 , 12 ]
Thread 1: [ 216 , 148 , 12 ]
Thread 1: [ 216 , 148 , 12 ]
Thread 1: [ 216 , 148 , 12 ]
Thread 1: [ 217 , 147 , 13 ]
Thread 1: [ 217 , 147 , 13 ]
Thread 1: [ 217 , 147 , 13 ]
Thread 1: [ 218 , 147 , 14 ]
Thread 1: [ 218 , 147 , 14 ]
```

```
Thread 1: [ 218 , 147 , 14 ]
Thread 1: [ 218 , 147 , 14 ]
Thread 1: [ 219 , 146 , 15 ]
Thread 1: [ 219 , 146 , 15 ]
Thread 1: [ 219 , 146 , 15 ]
Thread 1: [ 220 , 146 , 16 ]
Thread 1: [ 220 , 146 , 16 ]
Thread 1: [ 220 , 146 , 16 ]
Thread 1: [ 220 , 146 , 16 ]
Thread 1: [ 220 , 145 , 16 ]
Thread 1: [ 220 , 145 , 16 ]
Thread 1: [ 220 , 145 , 16 ]
Thread2-PQLEX: [ 220 , 146 , 16 ]
Thread 1: [ 220 , 145 , 16 ]
Thread 1: [ 221 , 145 , 17 ]
Thread 1: [ 221 , 145 , 17 ]
Thread3-PQEUC: [ 220 , 146 , 16 ]
Thread3-PQEUC: [ 221 , 145 , 17 ]
Thread 1: [ 221 , 145 , 17 ]
Thread4-PQBMX: [ 220 , 146 , 16 ]
Thread 1: [ 222 , 144 , 18 ]
Thread2-PQLEX: [ 222 , 144 , 18 ]
Thread4-PQBMX: [ 222 , 144 , 18 ]
Thread 1: [ 222 , 144 , 18 ]
Thread2-PQLEX: [ 221 , 145 , 17 ]
Thread3-PQEUC: [ 222 , 144 , 18 ]
Thread 1: [ 222 , 144 , 18 ]
Thread2-PQLEX: [ 222 , 144 , 18 ]
Thread4-PQBMX: [ 222 , 144 , 18 ]
Thread 1: [ 223 , 143 , 19 ]
```

```
Thread 1: [ 223 , 143 , 19 ]
Thread3-PQEUC: [ 223 , 143 , 19 ]
Thread3-PQEUC: [ 223 , 143 , 19 ]
Thread4-PQBMX: [ 222 , 144 , 18 ]
Thread 1: [ 223 , 143 , 19 ]
Thread3-PQEUC: [ 223 , 143 , 19 ]
Thread4-PQBMX: [ 220 , 146 , 16 ]
Thread2-PQLEX: [ 223 , 143 , 19 ]
Thread 1: [ 224 , 143 , 20 ]
Thread3-PQEUC: [ 224 , 143 , 20 ]
Thread2-PQLEX: [ 224 , 143 , 20 ]
Thread 1: [ 224 , 143 , 20 ]
Thread4-PQBMX: [ 224 , 143 , 20 ]
Thread 1: [ 224 , 143 , 20 ]
Thread2-PQLEX: [ 224 , 143 , 20 ]
Thread 1: [ 225 , 142 , 21 ]
Thread2-PQLEX: [ 224 , 143 , 20 ]
Thread3-PQEUC: [ 225 , 142 , 21 ]
Thread3-PQEUC: [ 224 , 143 , 20 ]
Thread4-PQBMX: [ 225 , 142 , 21 ]
Thread3-PQEUC: [ 224 , 143 , 20 ]
Thread 1: [ 225 , 142 , 21 ]
Thread4-PQBMX: [ 224 , 143 , 20 ]
Thread3-PQEUC: [ 225 , 142 , 21 ]
Thread 1: [ 225 , 142 , 21 ]
Thread4-PQBMX: [ 225 , 142 , 21 ]
Thread2-PQLEX: [ 225 , 142 , 21 ]
Thread4-PQBMX: [ 204 , 156 , 0 ]
Thread 1: [ 225 , 142 , 21 ]
Thread3-PQEUC: [ 225 , 142 , 21 ]
```

```
Thread3-PQEUC: [ 20 , 207 , 224 ]
Thread4-PQBMX: [ 11 , 215 , 209 ]
Thread3-PQEUC: [ 19 , 208 , 223 ]
Thread4-PQBMX: [ 19 , 208 , 223 ]
Thread2-PQLEX: [ 32 , 199 , 236 ]
Thread 1: [ 32 , 198 , 236 ]
Thread2-PQLEX: [ 32 , 198 , 236 ]
Thread3-PQEUC: [ 18 , 208 , 222 ]
Thread 1: [ 33 , 198 , 237 ]
Thread4-PQBMX: [ 22 , 206 , 226 ]
Thread2-PQLEX: [ 33 , 198 , 237 ]
Thread3-PQEUC: [ 12 , 213 , 216 ]
Thread3-PQEUC: [ 11 , 214 , 215 ]
Thread4-PQBMX: [ 27 , 202 , 231 ]
Thread 1: [ 33 , 198 , 237 ]
Thread4-PQBMX: [ 29 , 201 , 233 ]
Thread2-PQLEX: [ 33 , 198 , 237 ]
Thread2-PQLEX: [ 21 , 206 , 225 ]
Thread4-PQBMX: [ 30 , 200 , 234 ]
Thread 1: [ 33 , 198 , 237 ]
Thread3-PQEUC: [ 12 , 212 , 216 ]
Thread3-PQEUC: [ 12 , 212 , 216 ]
Thread2-PQLEX: [ 33 , 198 , 237 ]
Thread4-PQBMX: [ 30 , 200 , 234 ]
Thread3-PQEUC: [ 31 , 199 , 235 ]
Thread 1: [ 34 , 197 , 238 ]
Thread2-PQLEX: [ 34 , 197 , 238 ]
Thread3-PQEUC: [ 27 , 202 , 231 ]
Thread2-PQLEX: [ 30 , 200 , 234 ]
```

End of running:

```
Thread 1: [ 105 , 147 , 255 ]
Thread2-PQLEX: [ 105 , 147 , 255 ]
Thread3-PQEUC: [ 105 , 147 , 255 ]
Thread4-PQBMX: [ 89 , 158 , 255 ]
Thread 1: [ 105 , 147 , 255 ]
Thread2-PQLEX: [ 105 , 147 , 255 ]
Thread3-PQEUC: [ 105 , 147 , 255 ]
Thread 1: [ 105 , 147 , 255 ]
Thread4-PQBMX: [ 89 , 158 , 255 ]
Thread2-PQLEX: [ 105 , 147 , 255 ]
Thread3-PQEUC: [ 105 , 147 , 255 ]
Thread4-PQBMX: [ 102 , 149 , 255 ]
Thread 1: [ 105 , 147 , 255 ]
Thread2-PQLEX: [ 105 , 147 , 255 ]
Thread3-PQEUC: [ 105 , 147 , 255 ]
Thread 1: [ 105 , 147 , 255 ]
Thread4-PQBMX: [ 102 , 149 , 255 ]
Thread2-PQLEX: [ 105 , 147 , 255 ]
Thread 1: [ 105 , 147 , 255 ]
Thread3-PQEUC: [ 105 , 147 , 255 ]
Thread4-PQBMX: [ 102 , 149 , 255 ]
Thread3-PQEUC: [ 105 , 147 , 255 ]
Thread4-PQBMX: [ 104 , 148 , 255 ]
Thread2-PQLEX: [ 105 , 147 , 255 ]
Thread3-PQEUC: [ 100 , 150 , 255 ]
Thread2-PQLEX: [ 105 , 147 , 255 ]
Thread4-PQBMX: [ 105 , 147 , 255 ]

Process finished with exit code 0
```