Gebze Technical University
Computer Engineering

CSE 505 - 2018 Spring

HOMEWORK 5 REPORT

NURHAK ALTIN
1850044083

Course Assistant: Assoc. Prof. Dr. Erchan Aptoula

1 INTRODUCTION

1.1 Problem Definition

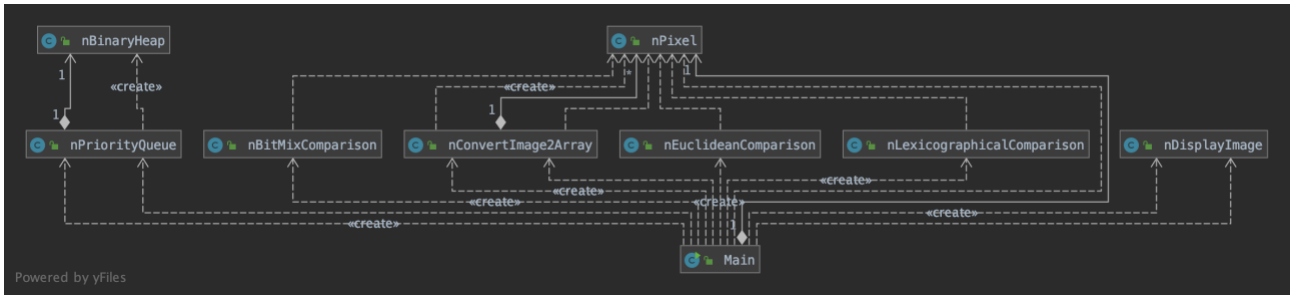
A rgb colored image has to be stored in the priority queue which uses the binary heap as an underlying data structure. An image consist os pixels and pixels represented as 3 dimesional vector which has red, green and the blue clor values as an unsigned integer. So the pixel values has to be inserted to priority queue with the different comparison methods which are lexicographical, Euclidean and the bitmix.

1.2 System Requirements

The program does not depens on the operating system that means all operating system are supported but the operating system has to be support the graphical user interface. Java 11 or higher versions, at least 4 GB memory including OS requirements.

2 METHOD

2.1 Class Diagrams



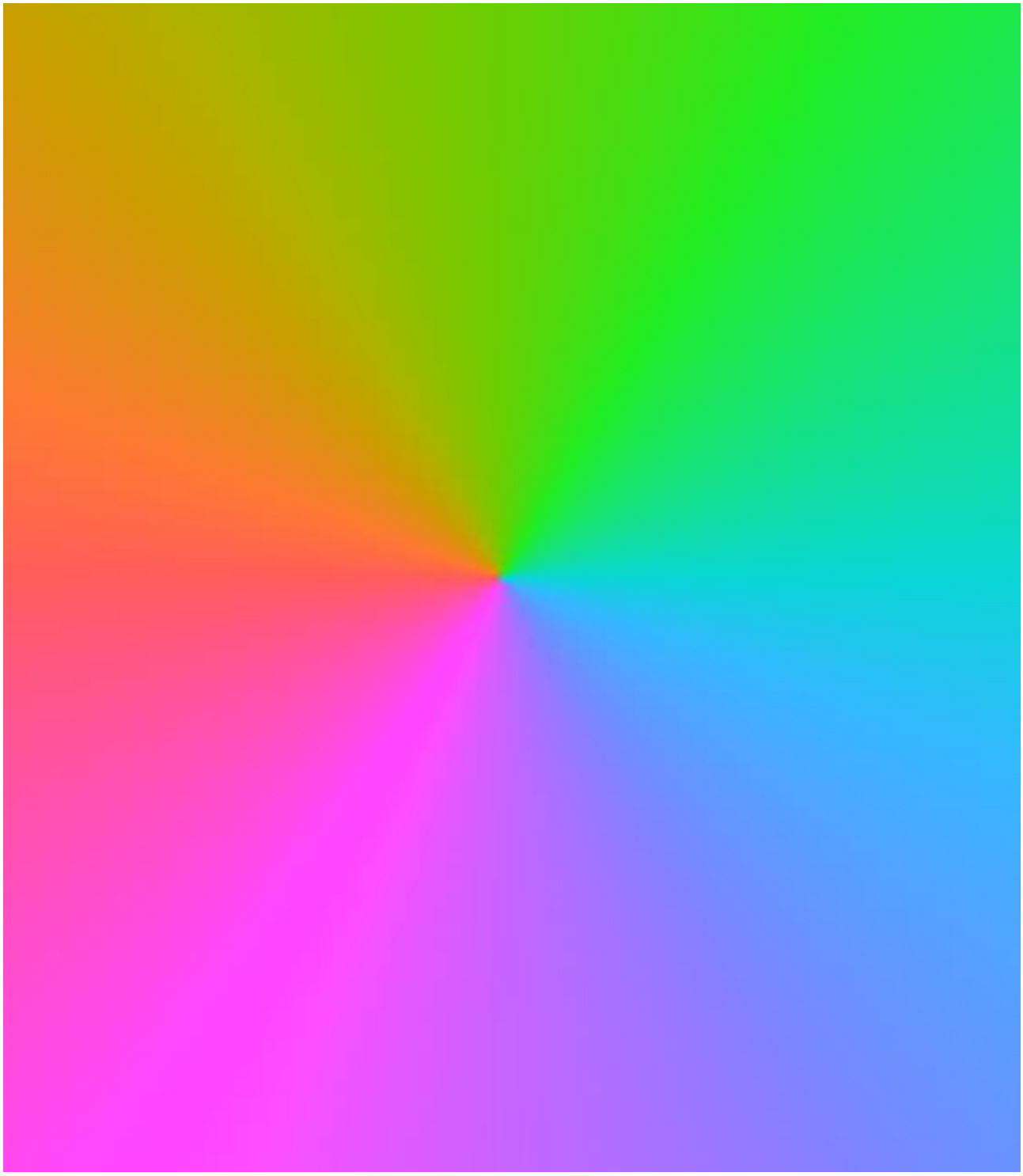
2.2 Problem Solution Approach

In the solution a binary heap implemented using arraylist data type and in binary heap class a comparator method does not implemented thats why if the user give the comparator method from outside the heap can add the element according to that comparrison method. A priority queue implemented using binary heap as an underlying data structure and the comparator same as the binary heap. 3 different comparison method implemented which are lexicographical, euclidean and the bitmix comparisons. The bitmix comparison gets the pixel values and concanate the bits from red, green and the blue one after the other and the final 24 bits integer value could be used in the comparison. The lexicographical comparison checks first the red value, then the green and the last blue value. The euclidean gets the norm of the vector and compare the objects according to their norm. Finally the main method reads an image from a file path that user gave and create a thread. That thread reads the values from the image an insert them to the 3 priority queue which has differen comparison methods. After 100 insertion the first thread create 3 other threads that remove from the priority queues and the display images.

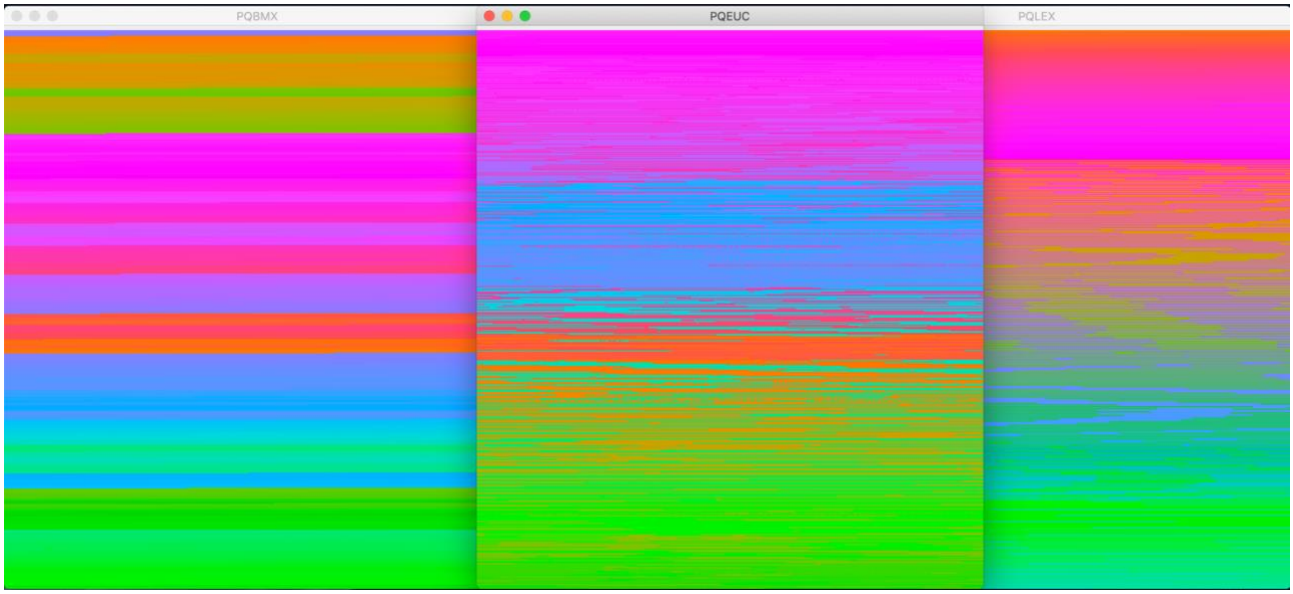
3 RESULT

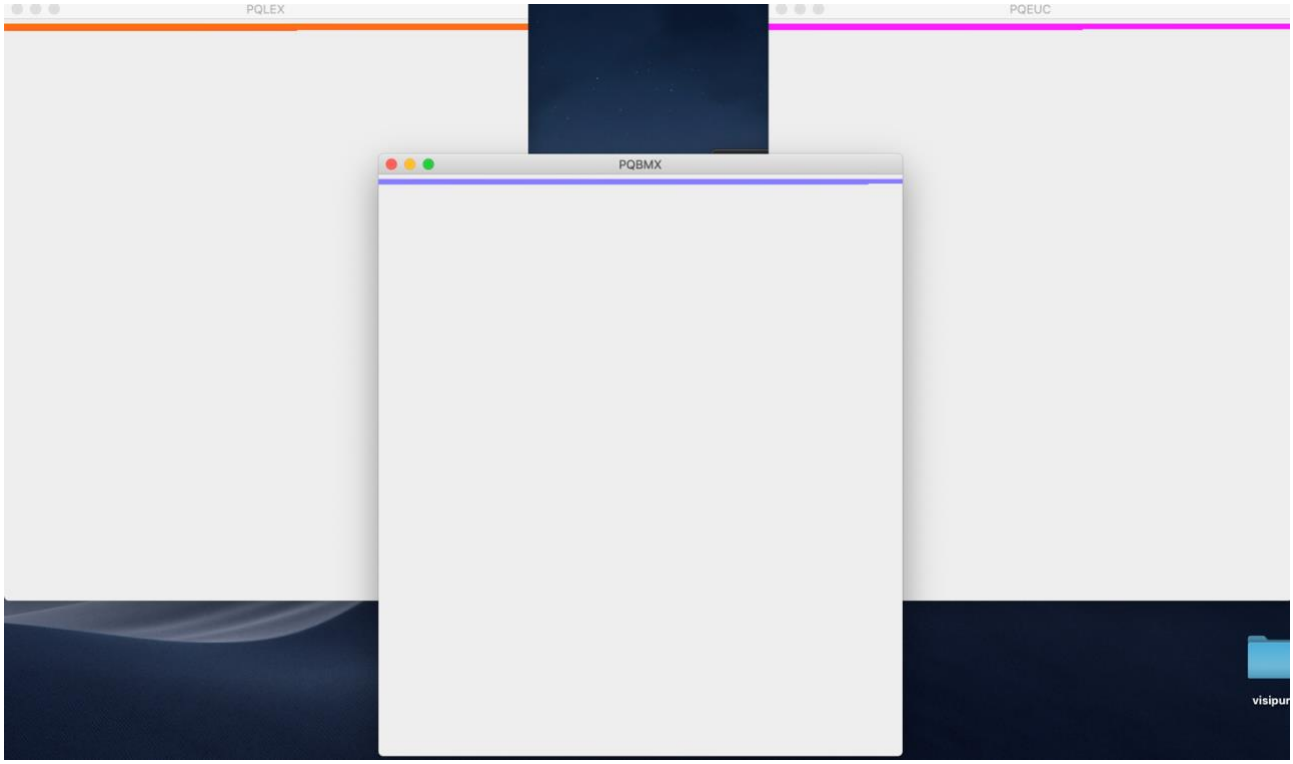
3.1 Test Cases

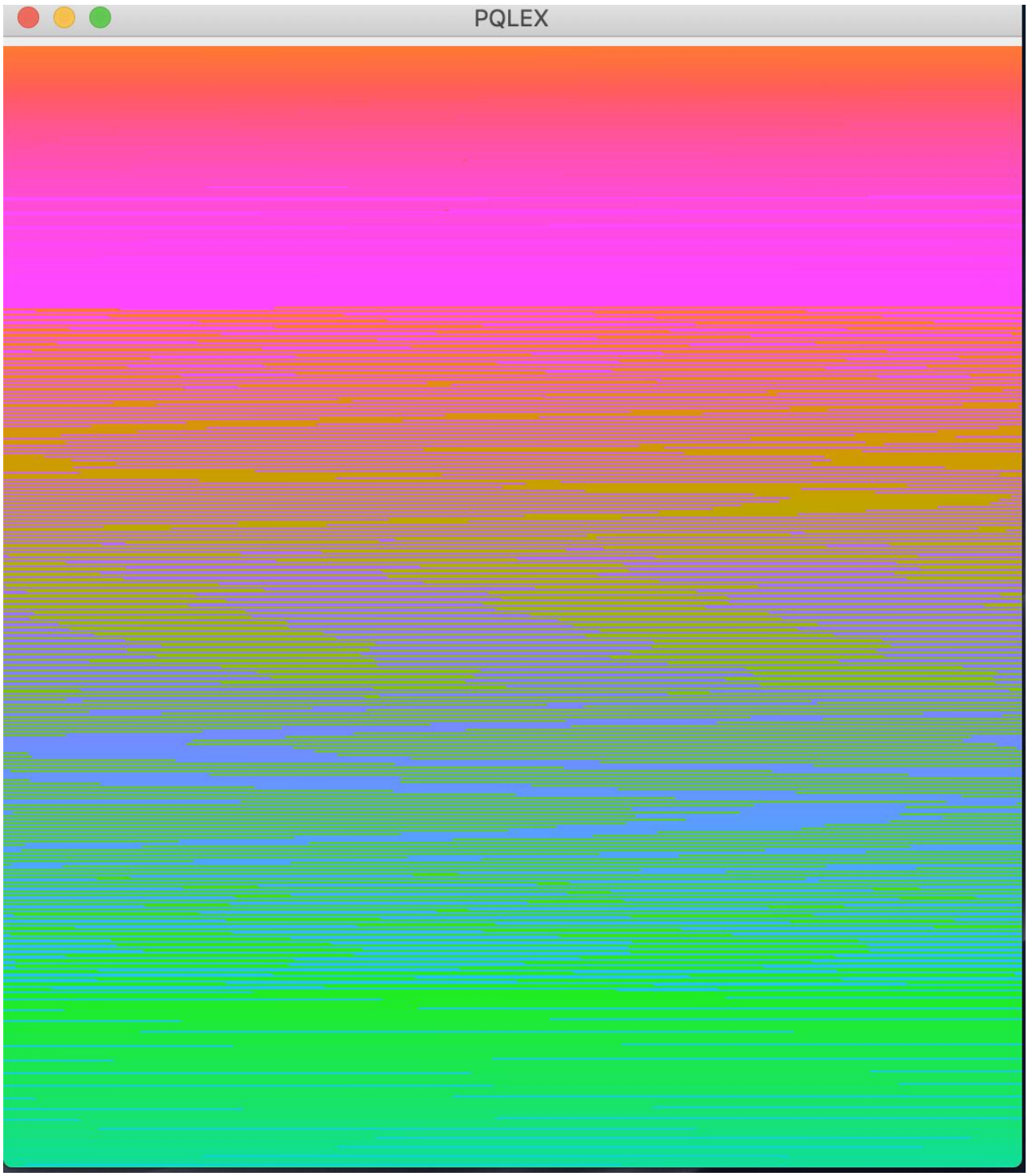
An colored image used to test the program which is the following.

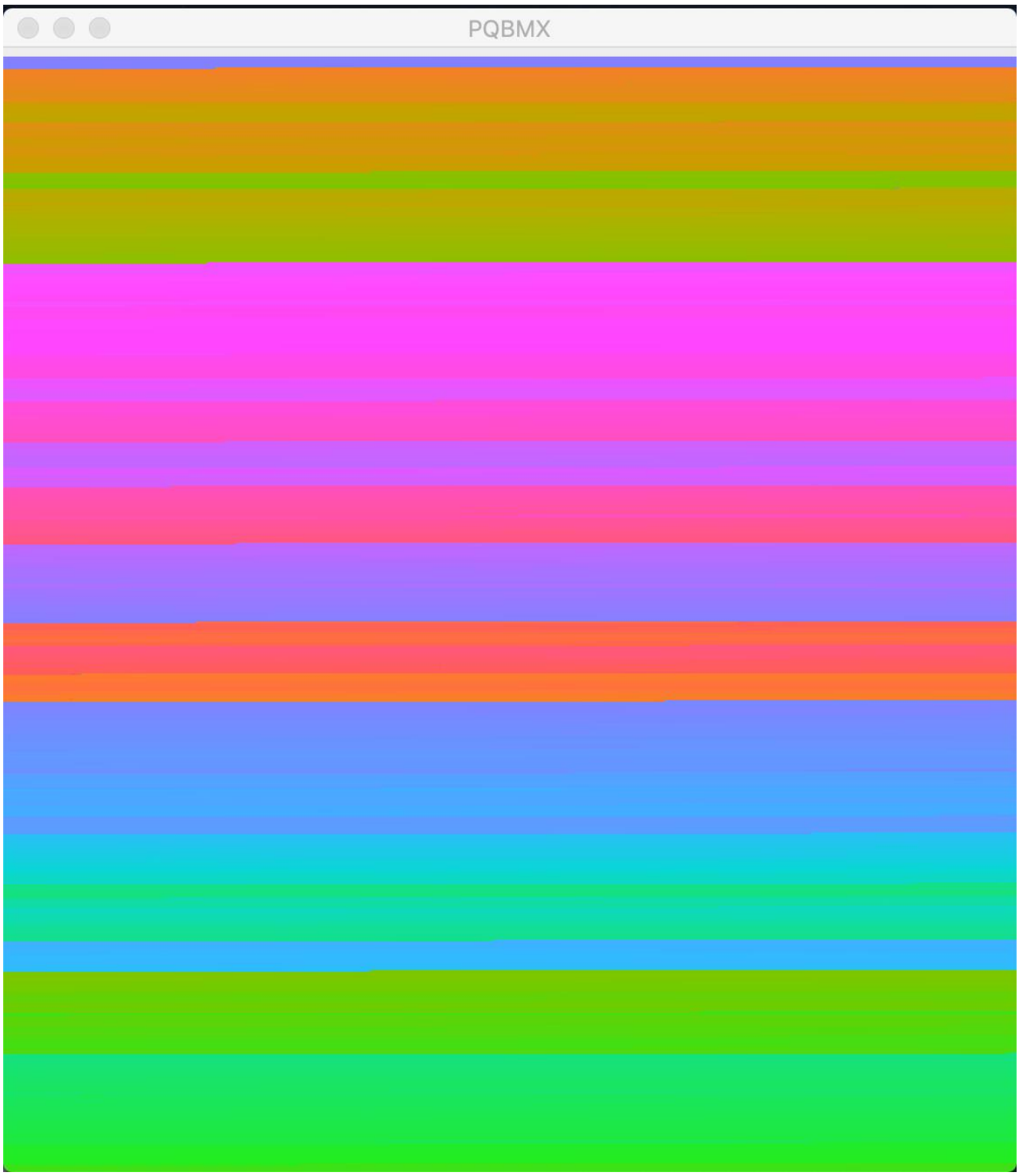


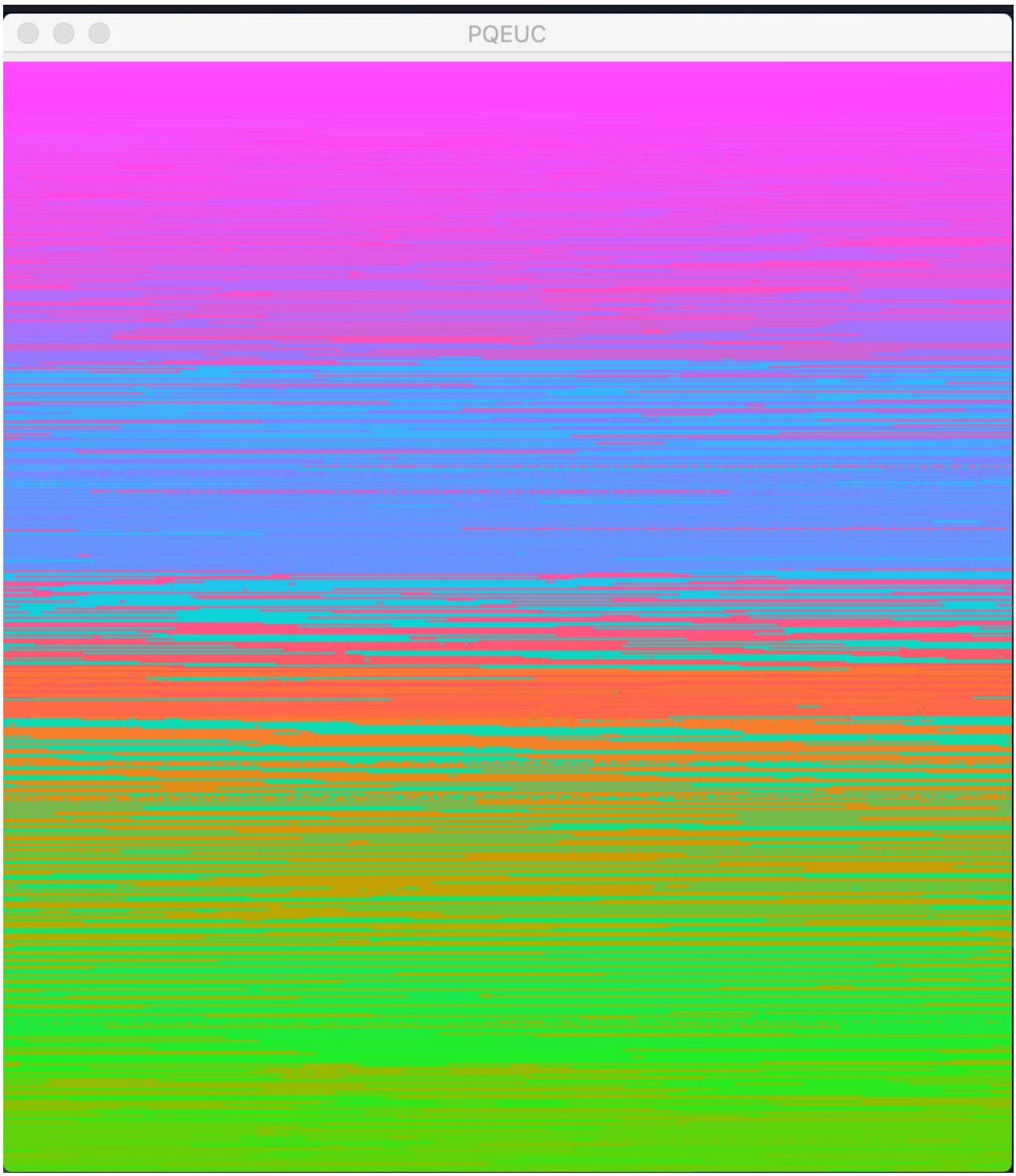
3.2 Running Results






PQLEX


PQBMX


PQEUC

```
Thread 1 PQEUC[inserting] :[ 255, 77, 208 ]
Thread 1 PQEUC[inserting] :[ 255, 77, 208 ]
Thread 1 PQEUC[inserting] :[ 255, 77, 209 ]
Thread 1 PQEUC[inserting] :[ 255, 77, 209 ]
Thread 1 PQEUC[inserting] :[ 255, 77, 209 ]
Thread 1 PQEUC[inserting] :[ 255, 77, 210 ]
Thread 1 PQEUC[inserting] :[ 255, 77, 210 ]
Thread 1 PQEUC[inserting] :[ 255, 77, 211 ]
Thread 1 PQEUC[inserting] :[ 255, 77, 211 ]
Thread 1 PQEUC[inserting] :[ 255, 77, 212 ]
Thread 1 PQEUC[inserting] :[ 255, 77, 212 ]
Thread 1 PQEUC[inserting] :[ 255, 77, 212 ]
Thread 1 PQEUC[inserting] :[ 255, 77, 212 ]
Thread 1 PQEUC[inserting] :[ 255, 76, 213 ]
Thread 1 PQEUC[inserting] :[ 255, 76, 213 ]
Thread 1 PQEUC[inserting] :[ 255, 76, 213 ]
Thread 1 PQEUC[inserting] :[ 255, 76, 214 ]
Thread 1 PQEUC[inserting] :[ 255, 76, 214 ]
Thread 18-PQLEX[removing] : [ 255, 118, 55 ]
Thread 20-PQBMX[removing] : [ 133, 128, 255 ]
Thread 1 PQEUC[inserting] :[ 255, 76, 214 ]
Thread 1 PQEUC[inserting] :[ 255, 76, 214 ]
Thread 1 PQEUC[inserting] :[ 255, 76, 215 ]
Thread 1 PQEUC[inserting] :[ 255, 76, 215 ]
Thread 1 PQEUC[inserting] :[ 255, 76, 215 ]
```