Project 1: Histogram (C++)

CSC 381 Rafael Carmilema

Due Date: Aug / 30 /2016

Algorithm Steps:

step 0: - open input and output files  
  
step 1: - read the image header from input file, the four numbers.  
            - dynamically allocate the histogram array  
         - initial the historgram array to 0  
  
step 2: // process the input file from left to right and top to bottom  
  
         p(i,j) <- read from input  
         hitogram[p(i,j)]++  
  
step 3: repeat step 2 until the file is empty  
  
step 4: output histogram array to output file  
  
step 5: close input file and output file

MAIN

import java.io.File;

import java.io.FileNotFoundException;

import java.io.PrintWriter;

import java.io.PrintStream;

import java.util.Scanner;

public class main {

public static void main(String[] args) {

// TODO Auto-generated method stub

if ( args.length < 2 ){ // the number of arguments should be 2

System.err.println("Invalid number of arguments.");

System.exit(1);

}

else {

File file = new File(args[0]);

try {

int count=0;

int row=0;

int col=0;

int min=0;

int max=0;

int[][] inputArray;

PrintStream outfile1 = new PrintStream(args[1]);

Scanner infile = new Scanner(file);

while (infile.hasNext()) {

count++;

if(count==1){

row=infile.nextInt();

System.out.println(row);

}

else if(count==2){

col=infile.nextInt();

System.out.println(col);

}

else if(count == 3){

min=infile.nextInt();

System.out.println(min);

}

else if(count ==4){

max=infile.nextInt();

System.out.println(max);

}

else{

break;

}

}// while

histogram test= new histogram(row, col, min,max);

inputArray = new int[row][col];

for(int r=0; r<row;r++){

for(int c=0; c<col;c++){

while(infile.hasNext()){

inputArray[r][c]=infile.nextInt();

test.computeHistogram(inputArray[r][c]);

}

}

}

test.print(outfile1, max);

infile.close();

outfile1.close();

}

catch(FileNotFoundException e){

System.out.println("File not Found");

}

}

}

}

Histogram Class

import java.io.PrintWriter;

import java.util.Scanner;

import java.io.PrintStream;

public class histogram {

int r;

int c;

int mi;

int ma;

int[] histo;

public histogram(int row,int col, int min, int max){

r=row;

c=col;

mi=min;

ma=max;

histo = new int[max+1];

for(int i=0; i<max; i++){

histo[i]=0;

}

}

public void computeHistogram(int index){

histo[index]++;

}

public void print(PrintStream outfile,int max){

for(int i=0; i<=max;i++){

if(histo[i]>=80){

outfile.println("(" + i + "):" + "80 +'s");

}

else{

outfile.print("(" + i + "):" + histo[i]);

for(int j=0; j<histo[i];j++){

outfile.print("+");

}

outfile.println();

}

}

}

}

(0):0

(1):80 +'s

(2):80 +'s

(3):80 +'s

(4):80 +'s

(5):80 +'s

(6):7+++++++

(7):6++++++

(8):35+++++++++++++++++++++++++++++++++++

(9):4++++

(10):5+++++

(11):7+++++++

(12):8++++++++

(13):6++++++

(14):9+++++++++

(15):3+++

(16):3+++

(17):0

(18):12++++++++++++

(19):1+

(20):3+++

(21):4++++

(22):7+++++++

(23):3+++

(24):7+++++++

(25):3+++

(26):0

(27):3+++

(28):15+++++++++++++++

(29):3+++

(30):7+++++++

(31):7+++++++

(32):7+++++++

(33):2++

(34):10++++++++++

(35):10++++++++++

(36):0

(37):0

(38):17+++++++++++++++++

(39):1+

(40):6++++++

(41):12++++++++++++

(42):10++++++++++

(43):15+++++++++++++++

(44):12++++++++++++

(45):6++++++

(46):2++

(47):2++

(48):80 +'s

(49):0

(50):0

(51):8++++++++

(52):2++

(53):1+

(54):2++

(55):11+++++++++++

(56):0

(57):0

(58):10++++++++++

(59):0

(60):8++++++++

(61):1+

(62):2++

(63):6++++++