CSCI 381 (Java)

Section 31

Project 7.1 Run Length Encoding

Essam Yousry

Due Date of soft copy: 04/11/2018

Due Date of hard copy: 04/12/2018

Algorithm Steps for Run Length Encoding (Method 3)

Step 0: r <-- 0

Step 1: Scan inFile left to right and top to bottom

c <-- 0

count <-- 0

currVal <-- p (r, c)

Step 2: output r and c and currVal to outFile

count ++

Step 3: c++

Step 4: nextVal <-- p (r, c) // read the next pixel

Step 5: if nextVal == currVal

count++

else

output count to outFile

currVal <-- nextVal

count <-- 1

output r and c and currVal

Step 6: repeat step 3 to step 5 until end of text line

Step 7: r++

Step 8: repeat Step 1 to Step 7 until end of file

Source Code

import java.util.Scanner;

import java.io.FileReader;

import java.io.IOException;

import java.io.FileWriter;

import java.io.File;

import java.io.FileOutputStream;

import java.io.PrintWriter;

public class RunLengthEncode{

    public static void main(String [] args){

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

        File outFile = new File (args[1]);

        int row, col, min, max;

        int [] vars = new int [4];

        try {

            Scanner sc = new Scanner(inFile);

            Scanner user = new Scanner(System.in);

            int c = 0;

            while (sc.hasNextInt() && c != 4){

                vars[c++] = sc.nextInt();

            }

            row = vars[0];

            col = vars[1];

            min = vars[2];

            max = vars[3];

            int [][] data = new int [row][col];

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

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

                    data[i][j] = sc.nextInt();

                }

            }

            int n;

            System.out.println("Please Specify Which Encoding Method You Want To Use (1 - 4)");

            System.out.println("1 - encode non-zeros without wrap around");

            System.out.println("2 - encode non-zeros with wrap around");

            System.out.println("3 - encode zeros and non-zeros without wrap around");

            System.out.println("4 - encode zeros and non-zeros with wrap around");

            System.out.print("Method Number: ");

            n = user.nextInt();

            if ((n < 1) || (n > 4)){

                System.out.println("The number should be between 1 and 4");

                System.exit(0);

            }

            switch (n){

                case 1: EncodeMethod1(data, row, col, min, max, outFile);

                    break;

                case 2: EncodeMethod2(data, row, col, min, max, outFile);

                    break;

                case 3: EncodeMethod3(data, row, col, min, max, outFile);

                    break;

                case 4: EncodeMethod4(data, row, col, min, max, outFile);

                    break;

            }

        }

        catch (Exception e) {

            System.out.println("Error " + e);

        }

    }

    public static boolean CheckforZeros(int [][] array, int col, int r){

        boolean check = false;

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

            if (array[r][j] != 0){

                check = true;

                break;

            }

        }

        return check;

    }

    public static int findNonZero(int [][] array, int col, int r){

        int c = 0;

        for(int j = 0; j < col; j++){

            if (array[r][j] != 0){

                c = j;

                break;

            }

        }

        return c;

    }

    //encode non-zeros without wrap around

    public static void EncodeMethod1(int [][] array, int row, int col, int min, int max, File outFile){

        int nextVal = 0;

        int currVal = 0;

        try{

            PrintWriter pw = new PrintWriter(new FileWriter(outFile));

            pw.print(row);

            pw.print(' ');

            pw.print(col);

            pw.print(' ');

            pw.print(min);

            pw.print(' ');

            pw.print(max);

            pw.print(' ');

            pw.println();

            pw.println("1");

            int count = 0;

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

                boolean check = CheckforZeros(array, col, i);

                if (check == true){

                    int c = findNonZero(array, col, i);

                    currVal = array[i][c];

                    pw.print(i + 1);

                    pw.print(' ');

                    pw.print(c + 1);

                    pw.print(' ');

                    pw.print(currVal);

                    pw.print(' ');

                }

                else continue;

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

                    if (array[i][j] != 0){

                        nextVal = array[i][j];

                        if (nextVal == currVal) count++;

                        else{

                            currVal = nextVal;

                            pw.println(count);

                            count = 1;

                            pw.print(i + 1);

                            pw.print(' ');

                            pw.print(j + 1);

                            pw.print(' ');

                            pw.print(currVal);

                            pw.print(' ');

                        }

                    }

                    if (j == col - 1){

                        pw.print(count);

                        pw.println();

                    }

                }

                count = 0;

            }

            pw.println();

            pw.println();

            pw.close();

        }

        catch (Exception e) {

            System.out.println("Error " + e);

        }

    }

    //encode non-zeros with wrap around

    public static void EncodeMethod2(int [][] array, int row, int col, int min, int max, File outFile){

        int nextVal = 0;

        int currVal = 0;

        int count = 0;

        try{

            PrintWriter pw = new PrintWriter(new FileWriter(outFile, true));

            pw.print(row);

            pw.print(' ');

            pw.print(col);

            pw.print(' ');

            pw.print(min);

            pw.print(' ');

            pw.print(max);

            pw.print(' ');

            pw.println();

            pw.println("2");

            int a = 0;

            int b = 0;

            while (b < col){

                if (array[a][b] != 0){

                    break;

                }

                else b++;

                if (b == col) a++;

            }

            int r = a;

            int c = b;

            currVal = array[r][c];

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

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

                    if (array[i][j] != 0){

                        nextVal = array[i][j];

                        if (nextVal == currVal) count++;

                        else{

                            currVal = nextVal;

                            if (count != 0) pw.println(count);

                            count = 1;

                            pw.print(i + 1);

                            pw.print(' ');

                            pw.print(j + 1);

                            pw.print(' ');

                            pw.print(currVal);

                            pw.print(' ');

                        }

                    }

                    else currVal = 0;

                }

            }

            loop1:

            for (int i = row - 1; i >= 0; i--){

                for (int j = col - 1; j >= 0; j--){

                    if (array[i][j] != 0){

                        currVal = array[i][j];

                        break loop1;

                    }

                }

            }

            int lastCount = 0;

            loop2:

            for (int i = row - 1; i >= 0; i--){

                for (int j = col - 1; j >= 0; j--){

                    if (array[i][j] != 0){

                        if (array[i][j] == currVal){

                            lastCount++;

                        }

                        else break loop2;

                    }

                }

            }

            pw.print(lastCount);

            pw.close();

        }

        catch (Exception e) {

            System.out.println("Error " + e);

        }

    }

    //encode zeros and non-zeros without wrap around

    public static void EncodeMethod3(int [][] array, int row, int col, int min, int max, File outFile){

        int r = 0;

        int nextVal = 0;

        int currVal = 0;

        int c = 0;

        int count = 0;

        try{

            PrintWriter pw = new PrintWriter(new FileWriter(outFile));

            pw.print(row);

            pw.print(' ');

            pw.print(col);

            pw.print(' ');

            pw.print(min);

            pw.print(' ');

            pw.print(max);

            pw.print(' ');

            pw.println();

            pw.println("3");

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

                currVal = array[i][0];

                pw.print(i + 1);

                pw.print(' ');

                pw.print(c + 1);

                pw.print(' ');

                pw.print(currVal);

                pw.print(' ');

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

                    nextVal = array[i][j];

                    if (nextVal == currVal) count++;

                    else{

                        currVal = nextVal;

                        pw.println(count);

                        count = 1;

                        pw.print(i + 1);

                        pw.print(' ');

                        pw.print(j + 1);

                        pw.print(' ');

                        pw.print(currVal);

                        pw.print(' ');

                    }

                    if (j == col - 1){

                        pw.print(count);

                        pw.println();

                    }

                }

                count = 0;

            }

            pw.close();

        }

        catch (Exception e) {

            System.out.println("Error " + e);

        }

    }

    //encode zeros and non-zeros with wrap around

    public static void EncodeMethod4(int [][] array, int row, int col, int min, int max, File outFile){

        int r = 0;

        int nextVal = 0;

        int currVal = 0;

        int c = 0;

        int count = 0;

        try{

            PrintWriter pw = new PrintWriter(new FileWriter(outFile));

            pw.print(row);

            pw.print(' ');

            pw.print(col);

            pw.print(' ');

            pw.print(min);

            pw.print(' ');

            pw.print(max);

            pw.print(' ');

            pw.println();

            pw.println("4");

            currVal = array[0][0];

            pw.print(r + 1);

            pw.print(' ');

            pw.print(c + 1);

            pw.print(' ');

            pw.print(currVal);

            pw.print(' ');

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

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

                    nextVal = array[i][j];

                    if (nextVal == currVal) count++;

                    else{

                        currVal = nextVal;

                        pw.println(count);

                        count = 1;

                        pw.print(i + 1);

                        pw.print(' ');

                        pw.print(j + 1);

                        pw.print(' ');

                        pw.print(currVal);

                        pw.print(' ');

                    }

                }

            }

            int lastCount = 0;

            loop:

            for (int i = row - 1; i >= 0; i--){

                for (int j = col - 1; j >= 0; j--){

                    if (array[i][j] == currVal){

                        lastCount++;

                    }

                    else break loop;

                }

            }

            pw.print(lastCount);

            pw.close();

        }

        catch (Exception e) {

            System.out.println("Error " + e);

        }

    }

}

Output







