**CS381-37: Project 7.2 (JAVA)**

**Yida Tao**

**Due date: Apr. 12, 2018**

Algorithm Steps:

step 0: inFile 🡨 open input file

outFile 🡨 open a text file to write

initializing those needed in the data structures

step 1: Read the image header from inFile

Write the header to outFile

step 2: methodUsed <-- read from inFile

if methodUsed is not within 1 – 4

exit with error message

step 3: case of methodUsed

1: call deCodeMethod1 (inFile) // on your own

2: call deCodeMethod2 (inFile) // on your own

3: call deCodeMethod3 (inFile) // on your own

4: call deCodeMethod4 (inFile)) // on your own

step 4: closed all files

**Source Code**

**public** **class** project\_7\_2 {

**public** **static** **void** main(String[] args) {

**try**{

System.***out***.println("argument 1:" + args[0]);

System.***out***.println("argument 2:" + args[1]);

}**catch**(ArrayIndexOutOfBoundsException e){

System.***out***.println("no arguments");

System.*exit*(0);

}

System.***out***.println("Method 1) Decode without zero and no wrap-around");

System.***out***.println("Method 2) Decode without zero and wrap-around");

System.***out***.println("Method 3) Decode with zero and no wrap-around");

System.***out***.println("Method 4) Decode with zero and wrap-around");

**new** runLengthDecoder(args[0], args[1]).decode();

}

}

**import** java.io.\*;

**import** java.util.Scanner;

**public** **class** runLengthDecoder {

**int** method;

String infile;

String outfile;

**int** numRows = 0;

**int** numCols = 0;

**int** minVal = 0;

**int** maxVal = 0;

**public** runLengthDecoder(String in, String out) {

infile = in;

outfile = out;

}

**public** **void** decode() {

Scanner sc = **null**;

FileOutputStream fos = **null**;

**int** srow = 0;

**int** scol = 0;

**int** greyScale = 0;

**int** length = 0;

**int** totalCount = 0;

**try** {

sc = **new** Scanner(**new** File(infile));

} **catch** (FileNotFoundException e) {

e.printStackTrace();

System.***out***.println("Cant find the file: " + infile);

}

numRows = sc.nextInt();

numCols = sc.nextInt();

minVal = sc.nextInt();

maxVal = sc.nextInt();

method = sc.nextInt();

**if**(method < 1 || method > 4){

System.***out***.println("invaild method number");

System.*exit*(0);

}

**try** {

fos = **new** FileOutputStream(outfile);

System.*setOut*(**new** PrintStream(fos));

} **catch** (FileNotFoundException e) {

e.printStackTrace();

}

System.***out***.println(numRows + " " + numCols + " " + minVal + " " + maxVal);

**if**(method == 1){

**while**(sc.hasNextInt()){

srow = sc.nextInt();

scol = sc.nextInt();

greyScale = sc.nextInt();

length = sc.nextInt();

**while**(totalCount < (srow\*numCols + scol)){

System.***out***.print(0 + " ");

totalCount ++;

**if**(totalCount % numCols == 0){

System.***out***.println();

}

}

**while**(length > 0){

System.***out***.print(greyScale + " ");

totalCount ++;

length --;

}

**if**(totalCount % numCols == 0){

System.***out***.println();

}

}

**while**(totalCount < numCols\*numRows){

System.***out***.print(0 + " ");

totalCount ++;

}

}

**else** **if**(method == 2){

**while**(sc.hasNextInt()){

srow = sc.nextInt();

scol = sc.nextInt();

greyScale = sc.nextInt();

length = sc.nextInt();

**while**(totalCount < (srow\*numCols + scol)){

System.***out***.print(0 + " ");

totalCount ++;

**if**(totalCount % numCols == 0){

System.***out***.println();

}

}

**while**(length > 0){

System.***out***.print(greyScale + " ");

totalCount ++;

**if**(totalCount % numCols == 0){

System.***out***.println();

}

length --;

}

}

**while**(totalCount < numCols\*numRows){

System.***out***.print(0 + " ");

totalCount ++;

}

}

**else** **if**(method == 3){

**while**(sc.hasNextInt()){

srow = sc.nextInt();

scol = sc.nextInt();

greyScale = sc.nextInt();

length = sc.nextInt();

**while**(length > 0){

System.***out***.print(greyScale + " ");

totalCount ++;

length --;

}

**if**(totalCount % numCols == 0){

System.***out***.println();

}

}

}

**else** **if**(method == 4){

**while**(sc.hasNextInt()){

srow = sc.nextInt();

scol = sc.nextInt();

greyScale = sc.nextInt();

length = sc.nextInt();

**while**(length > 0){

System.***out***.print(greyScale + " ");

totalCount ++;

**if**(totalCount % numCols == 0){

System.***out***.println();

}

length --;

}

}

}

sc.close();

**try** {

fos.close();

System.*setOut*(**new** PrintStream(**new** FileOutputStream(FileDescriptor.***out***)));

} **catch** (IOException e) {

e.printStackTrace();

}

System.***out***.println("Done");

}

}

**Input 1**

15 20 0 9

1

0 15 4 5

1 2 4 9

2 5 3 15

3 0 3 3

3 5 3 6

3 11 7 9

4 0 7 10

5 2 1 8

5 13 8 4

5 17 9 3

6 0 9 5

6 8 9 5

6 17 8 3

7 0 8 3

7 3 9 2

7 8 9 5

8 7 2 5

8 12 3 2

8 14 4 6

12 2 1 18

13 0 1 10

13 10 6 10

14 0 6 10

**Input 2**

15 20 0 9

2

0 15 4 5

1 2 4 9

2 5 3 18

3 5 3 6

3 11 7 19

5 2 1 8

5 13 8 4

5 17 9 8

6 8 9 5

6 17 8 6

7 3 9 2

7 8 9 5

8 7 2 5

8 12 3 2

8 14 4 6

12 2 1 28

13 10 6 20

**Input 3**

15 20 0 9

3

0 0 0 15

0 15 4 5

1 0 0 2

1 2 4 9

1 11 0 9

2 0 0 5

2 5 3 15

3 0 3 3

3 3 0 2

3 5 3 6

3 11 7 9

4 0 7 10

4 10 0 10

5 0 0 2

5 2 1 8

5 10 0 3

5 13 8 4

5 17 9 3

6 0 9 5

6 5 0 3

6 8 9 5

6 13 0 4

6 17 8 3

7 0 8 3

7 3 9 2

7 5 0 3

7 8 9 5

7 13 0 7

8 0 0 7

8 7 2 5

8 12 3 2

8 14 4 6

9 0 0 20

10 0 0 20

11 0 0 20

12 0 0 2

12 2 1 18

13 0 1 10

13 10 6 10

14 0 6 10

14 10 0 10

**Input 4**

15 20 0 9

4

0 0 0 15

0 15 4 5

1 0 0 2

1 2 4 9

1 11 0 14

2 5 3 18

3 3 0 2

3 5 3 6

3 11 7 19

4 10 0 12

5 2 1 8

5 10 0 3

5 13 8 4

5 17 9 8

6 5 0 3

6 8 9 5

6 13 0 4

6 17 8 6

7 3 9 2

7 5 0 3

7 8 9 5

7 13 0 14

8 7 2 5

8 12 3 2

8 14 4 6

9 0 0 62

12 2 1 28

13 10 6 20

14 10 0 10

**Output**

15 20 0 9

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 4 4 4 4 4

0 0 4 4 4 4 4 4 4 4 4 0 0 0 0 0 0 0 0 0

0 0 0 0 0 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3

3 3 3 0 0 3 3 3 3 3 3 7 7 7 7 7 7 7 7 7

7 7 7 7 7 7 7 7 7 7 0 0 0 0 0 0 0 0 0 0

0 0 1 1 1 1 1 1 1 1 0 0 0 8 8 8 8 9 9 9

9 9 9 9 9 0 0 0 9 9 9 9 9 0 0 0 0 8 8 8

8 8 8 9 9 0 0 0 9 9 9 9 9 0 0 0 0 0 0 0

0 0 0 0 0 0 0 2 2 2 2 2 3 3 4 4 4 4 4 4

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1 1 6 6 6 6 6 6 6 6 6 6

6 6 6 6 6 6 6 6 6 6 0 0 0 0 0 0 0 0 0 0