**CSC 381Image Processing (CPP)**

**Project: 8 Distance Transform**

**Student: Jeffrey Pulinat**

**Language: C++**

**Due date: C++ soft copy: 2/23/2018 Friday before midnight**

**+1 pt for early submission, deadline: 2/21/2018 Wednesday before midnight**

**Due date: C++ hard copy: 2/27/2017 Tuesday in class before exam**

step 0: read the image header

dynamically allocate zerpFramedAry with extra 2 rows and 2 cols

step 1: zeroFrame the zerpFramedAry.

Step 2: loadImage

step 3: firstPassDistance (…) // 8-distance algorithm taught in class

step 4: prettyPrintof the result of Pass-1 to outFile2

with caption indicating the result of pass-1,

step 5: secondPassDistance (…) // 8-distance algorithm taught in class

// In second pass, you need to keep track the newMinVal and newMaxVal

Step 6: prettyPrintDistace of the result of Pass-2 to outFile2 w/ caption

Step 7: Write the result of Pass-2 (without the 2 extra rows and columns) to outFile1 with updated image header.

Step 8: close all files.

**B) Source Code:**

#include <iostream>

#include <fstream>

#include <string>

#include <stdlib.h>

**using** **namespace** std**;**

static int numRows**;**

static int numCols**;**

static int minVal**;**

static int maxVal**;**

static int newMinVal**;**

static int newMaxVal**;**

static int neighborAry **[**5**];**

static int**\*\*** zeroFramedAry**;**

void zeroFramed**(**ifstream**&** input**){**

**for(**int col**=**0**;** col**<=**numCols**+**1**;** col**++){**

zeroFramedAry**[**0**][**col**]** **=** 0**;** //top

zeroFramedAry**[**numRows**+**1**][**col**]** **=** 0**;** //bottom

**}**

**for(**int row**=**0**;** row**<=**numRows**+**1**;** row**++){**

zeroFramedAry**[**row**][**0**]** **=** 0**;** //left side

zeroFramedAry**[**row**][**numCols**+**1**]** **=** 0**;** //right side

**}**

**}**

void loadImage**(**ifstream**&** input**){**

int param**;**

input **>>** param**;**

numRows **=** param**;**

input **>>** param**;**

numCols **=** param**;**

input **>>** param**;**

minVal **=** param**;**

input **>>** param**;**

maxVal **=** param**;**

zeroFramedAry **=** **new** int **\*[**numRows**+**2**];**

**for(**int i**=**0**;** i**<**numRows**+**2**;** **++**i**){**

zeroFramedAry**[**i**]** **=** **new** int**[**numCols**+**2**];**

**}**

**for(**int row **=** 1**;** row**<=** numRows**;** row**++){**

**for(**int col **=** 1**;** col**<=** numCols**;** col**++){**

int value**;**

input **>>** value**;**

zeroFramedAry**[**row**][**col**]** **=** value**;**

**}**

**}**

**}**

void loadNeighbors**(**int row**,** int col**,** int pass**)** **{** //load 3x3 neighbors

/\* a|b|c

\* d|e|f e = p(i,j)

\* g|h|i \*/

**if(**pass**==**1**){**

neighborAry**[**0**]=**zeroFramedAry**[**row**-**1**][**col**-**1**];** //a

neighborAry**[**1**]=**zeroFramedAry**[**row**-**1**][**col**];** //b

neighborAry**[**2**]=**zeroFramedAry**[**row**-**1**][**col**+**1**];** //c

neighborAry**[**3**]=**zeroFramedAry**[**row**][**col**-**1**];** //d

neighborAry**[**4**]=**zeroFramedAry**[**row**][**col**];** //e p(i,j)

//System.out.println(Arrays.toString(neighborAry));

**}**

**if(**pass**==**2**){**

neighborAry**[**4**]=**zeroFramedAry**[**row**][**col**];** //e p(i,j)

neighborAry**[**0**]=**zeroFramedAry**[**row**][**col**+**1**];** //f

neighborAry**[**1**]=**zeroFramedAry**[**row**+**1**][**col**-**1**];** //g

neighborAry**[**2**]=**zeroFramedAry**[**row**+**1**][**col**];** //h

neighborAry**[**3**]=**zeroFramedAry**[**row**+**1**][**col**+**1**];** //i

//System.out.println(Arrays.toString(neighborAry));

**}**

**}**

void firstPassDistance**()** **{**

int min**;**

int max**=**0**;**

bool anyzero **=false;**

**for(**int row **=** 1**;** row**<=**numRows**;** row**++){**

**for(**int col **=** 1**;** col**<=**numCols**;** col**++){**

**if(**zeroFramedAry**[**row**][**col**]>**0**){** //p(i,j) is not 0

loadNeighbors**(**row**,**col**,**1**);**

**if(**numRows**>**numCols**)**min **=** numRows**;** //assign to highest possible value aka numRows or numCols

**if(**numCols**>**numRows**)**min **=** numCols**;**

**for(**int i**=**0**;** i**<=**3**;** i**++)** **{** //doesnt compare p(i,j)

**if(**neighborAry**[**i**]<**min**)** **{**

min **=** neighborAry**[**i**];**

**}**

**}**

min**++;**

zeroFramedAry**[**row**][**col**]=**min**;**

**if(**min**>**max**)**max**=**min**;**

**}else** anyzero**=true;** //if a zero exists

**}**

**}**

**if(**anyzero**==false)**newMinVal**=**1**;**

**else** newMinVal**=**0**;**

newMaxVal**=**max**;**

**}**

void secondPassDistance**()** **{**

int min**;**

int max**=**0**;**

bool anyzero **=false;**

**for(**int row **=** numRows**;** row**>=**1**;** row**--){**

**for(**int col **=** numCols**;** col**>=**1**;** col**--){**

**if(**zeroFramedAry**[**row**][**col**]>**0**){** //p(i,j) is not 0

loadNeighbors**(**row**,**col**,**2**);**

**if(**numRows**>**numCols**)**min **=** numRows**;** //assign to highest possible value aka numRows or numCols

**if(**numCols**>**numRows**)**min **=** numCols**;**

**for(**int i**=**0**;** i**<=**3**;** i**++)** **{** //doesnt compare p(i,j)

**if(**neighborAry**[**i**]<**min**)** **{**

min **=** neighborAry**[**i**];**

**}**

**}**

min**++;**

**if(**neighborAry**[**4**]<**min**)**min**=**neighborAry**[**4**];** //compare min(min neighbor +1 and p(i,j))

zeroFramedAry**[**row**][**col**]=**min**;**

**if(**min**>**max**)**max**=**min**;**

**}else** anyzero**=true;** //if a zero exists

**}**

**}**

**if(**anyzero**==false)**newMinVal**=**1**;**

**else** newMinVal**=**0**;**

newMaxVal**=**max**;**

**}**

void prettyPrint**(**ifstream**&** input**,** ofstream**&** output**,** bool zborder**,** bool iheader**){**

**if(**iheader**==true)**output**<<**endl**<<**numRows**<<**" "**<<**numCols**<<**" "**<<**newMinVal**<<**" "**<<**newMaxVal**;** //prints image params

output**<<**endl**;**

**if(**zborder**==false){**

**for(**int row **=** 1**;** row**<=**numRows**;** row**++){** //print out from [1][1] w/o border

**for(**int col **=** 1**;** col**<=**numCols**;** col**++){**

int pixel\_val **=** zeroFramedAry**[**row**][**col**];**

**if(**pixel\_val **>** 0**)** output **<<** pixel\_val **<<**" "**;** //if not 0. output 1 space,

**else** output **<<** " "**;** // if 0. output 2 space

**}**

output**<<**endl**;**

**}**

**}**

**if(**zborder**==true){**

**for(**int row **=** 0**;** row**<=**numRows**+**1**;** row**++){** //print out from [0][0] with border

**for(**int col **=** 0**;** col**<=**numCols**+**1**;** col**++){**

int pixel\_val**=** zeroFramedAry**[**row**][**col**];**

**if(**pixel\_val **>** 0**)** output **<<** pixel\_val **<<**" "**;** //if not 0. output 1 space,

**else** output **<<** " "**;** // if 0. output 2 space

**}**

output**<<**endl**;**

**}**

**}**

**}**

int main**(**int argc**,** char **\***argv**[])** **{**

ifstream input**(**argv**[**1**]);**

string arg1 **=**argv**[**1**];**

ofstream output2**(**argv**[**3**]);** //outFile2

loadImage**(**input**);**

zeroFramed**(**input**);**

output2**<<**"PASS 1"**;**

firstPassDistance**();**

prettyPrint**(**input**,**output2**,false,false);**

secondPassDistance**();**

output2**<<**endl**<<**"------------"**<<**endl**<<**"PASS 2"**;**

prettyPrint**(**input**,**output2**,false,false);**

output2**.**close**();**

ofstream output1**(**argv**[**2**]);** //outFile1

output1**<<**"Pass 2"**<<**endl**;**

prettyPrint**(**input**,**output1**,false,true);** //Pass 2 values

input**.**close**();**

output1**.**close**();**

**for** **(**int i **=** 0**;** i **<** numRows**;** **++**i **)delete** **[]** zeroFramedAry**[**i**];**

**delete** **[]** zeroFramedAry**;**

**}**

**OUTPUT FILES**

**Data1outFile1.txt**

Pass 2

15 19 1 8

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1

1 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 1

1 2 3 4 4 4 4 4 4 4 4 4 4 4 4 4 3 2 1

1 2 3 4 5 5 5 5 5 5 5 5 5 5 5 4 3 2 1

1 2 3 4 5 6 6 6 6 6 6 6 6 6 5 4 3 2 1

1 2 3 4 5 6 7 7 7 7 7 7 7 6 5 4 3 2 1

1 2 3 4 5 6 7 8 8 8 8 8 7 6 5 4 3 2 1

1 2 3 4 5 6 7 7 7 7 7 7 7 6 5 4 3 2 1

1 2 3 4 5 6 6 6 6 6 6 6 6 6 5 4 3 2 1

1 2 3 4 5 5 5 5 5 5 5 5 5 5 5 4 3 2 1

1 2 3 4 4 4 4 4 4 4 4 4 4 4 4 4 3 2 1

1 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 1

1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

**Data1outFile2.txt**

PASS 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1

1 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 1

1 2 3 4 4 4 4 4 4 4 4 4 4 4 4 4 3 2 1

1 2 3 4 5 5 5 5 5 5 5 5 5 5 5 4 3 2 1

1 2 3 4 5 6 6 6 6 6 6 6 6 6 5 4 3 2 1

1 2 3 4 5 6 7 7 7 7 7 7 7 6 5 4 3 2 1

1 2 3 4 5 6 7 8 8 8 8 8 7 6 5 4 3 2 1

1 2 3 4 5 6 7 8 9 9 9 8 7 6 5 4 3 2 1

1 2 3 4 5 6 7 8 9 10 9 8 7 6 5 4 3 2 1

1 2 3 4 5 6 7 8 9 10 9 8 7 6 5 4 3 2 1

1 2 3 4 5 6 7 8 9 10 9 8 7 6 5 4 3 2 1

1 2 3 4 5 6 7 8 9 10 9 8 7 6 5 4 3 2 1

1 2 3 4 5 6 7 8 9 10 9 8 7 6 5 4 3 2 1

1 2 3 4 5 6 7 8 9 10 9 8 7 6 5 4 3 2 1

------------

PASS 2

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1

1 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 1

1 2 3 4 4 4 4 4 4 4 4 4 4 4 4 4 3 2 1

1 2 3 4 5 5 5 5 5 5 5 5 5 5 5 4 3 2 1

1 2 3 4 5 6 6 6 6 6 6 6 6 6 5 4 3 2 1

1 2 3 4 5 6 7 7 7 7 7 7 7 6 5 4 3 2 1

1 2 3 4 5 6 7 8 8 8 8 8 7 6 5 4 3 2 1

1 2 3 4 5 6 7 7 7 7 7 7 7 6 5 4 3 2 1

1 2 3 4 5 6 6 6 6 6 6 6 6 6 5 4 3 2 1

1 2 3 4 5 5 5 5 5 5 5 5 5 5 5 4 3 2 1

1 2 3 4 4 4 4 4 4 4 4 4 4 4 4 4 3 2 1

1 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 1

1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

**Data2Outfile1.txt**

Pass 2

20 19 0 4

1

1 1 1

1 1 2 1 1

1 1 2 2 2 1 1

1 1 2 2 3 2 2 1 1

1 1 2 2 3 3 3 2 2 1 1

1 1 2 2 3 3 4 3 3 2 2 1 1

1 1 2 2 3 3 3 3 3 3 3 2 2 1 1

1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 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 2 1 1

1 1 2 2 2 1 1

1 1 2 2 3 2 2 1 1

1 1 2 2 3 3 3 2 2 1 1

1 1 2 2 3 3 4 3 3 2 2 1 1

1 1 2 2 3 3 3 3 3 3 3 2 2 1 1

1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

**Data2Outfile2.txt**

PASS 1

1

1 1 1

1 1 2 1 1

1 1 2 2 2 1 1

1 1 2 2 3 2 2 1 1

1 1 2 2 3 3 3 2 2 1 1

1 1 2 2 3 3 4 3 3 2 2 1 1

1 1 2 2 3 3 4 4 4 3 3 2 2 1 1

1 1 2 2 3 3 4 4 5 4 4 3 3 2 2 1 1

1 1 2 2 3 3 4 4 5 5 5 4 4 3 3 2 2 1 1

1

1 1 1

1 1 2 1 1

1 1 2 2 2 1 1

1 1 2 2 3 2 2 1 1

1 1 2 2 3 3 3 2 2 1 1

1 1 2 2 3 3 4 3 3 2 2 1 1

1 1 2 2 3 3 4 4 4 3 3 2 2 1 1

1 1 2 2 3 3 4 4 5 4 4 3 3 2 2 1 1

1 1 2 2 3 3 4 4 5 5 5 4 4 3 3 2 2 1 1

------------

PASS 2

1

1 1 1

1 1 2 1 1

1 1 2 2 2 1 1

1 1 2 2 3 2 2 1 1

1 1 2 2 3 3 3 2 2 1 1

1 1 2 2 3 3 4 3 3 2 2 1 1

1 1 2 2 3 3 3 3 3 3 3 2 2 1 1

1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 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 2 1 1

1 1 2 2 2 1 1

1 1 2 2 3 2 2 1 1

1 1 2 2 3 3 3 2 2 1 1

1 1 2 2 3 3 4 3 3 2 2 1 1

1 1 2 2 3 3 3 3 3 3 3 2 2 1 1

1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

**Data3Outfile1.txt**

Pass 2

17 17 0 5

1

1 1 1

1 1 2 1 1

1 1 2 2 2 1 1

1 1 2 2 3 2 2 1 1

1 1 2 2 3 3 3 2 2 1 1

1 1 2 2 3 3 4 3 3 2 2 1 1

1 1 2 2 3 3 4 4 4 3 3 2 2 1 1

1 1 2 2 3 3 4 4 5 4 4 3 3 2 2 1 1

1 1 2 2 3 3 4 4 4 3 3 2 2 1 1

1 1 2 2 3 3 4 3 3 2 2 1 1

1 1 2 2 3 3 3 2 2 1 1

1 1 2 2 3 2 2 1 1

1 1 2 2 2 1 1

1 1 2 1 1

1 1 1

1

**Data3Outfile2.txt**

PASS 1

1

1 1 1

1 1 2 1 1

1 1 2 2 2 1 1

1 1 2 2 3 2 2 1 1

1 1 2 2 3 3 3 2 2 1 1

1 1 2 2 3 3 4 3 3 2 2 1 1

1 1 2 2 3 3 4 4 4 3 3 2 2 1 1

1 1 2 2 3 3 4 4 5 4 4 3 3 2 2 1 1

1 2 3 3 4 4 5 5 5 4 4 3 3 2 2

1 2 3 4 5 5 6 5 5 4 4 3 3

1 2 3 4 5 6 6 5 5 4 4

1 2 3 4 5 6 6 5 5

1 2 3 4 5 6 6

1 2 3 4 5

1 2 3

1

------------

PASS 2

1

1 1 1

1 1 2 1 1

1 1 2 2 2 1 1

1 1 2 2 3 2 2 1 1

1 1 2 2 3 3 3 2 2 1 1

1 1 2 2 3 3 4 3 3 2 2 1 1

1 1 2 2 3 3 4 4 4 3 3 2 2 1 1

1 1 2 2 3 3 4 4 5 4 4 3 3 2 2 1 1

1 1 2 2 3 3 4 4 4 3 3 2 2 1 1

1 1 2 2 3 3 4 3 3 2 2 1 1

1 1 2 2 3 3 3 2 2 1 1

1 1 2 2 3 2 2 1 1

1 1 2 2 2 1 1

1 1 2 1 1

1 1 1

1

**Data4Outfile1.txt**

Pass 2

17 17 0 5

1

1 1 1

1 1 2 1 1

1 1 2 2 2 1 1

1 1 2 2 3 2 2 1 1

1 1 2 2 3 3 3 2 2 1 1

1 2 2 3 3 4 3 3 2 2 1

1 2 3 3 4 4 4 3 3 2 1

1 2 3 4 4 5 4 4 3 2 1

1 2 3 3 4 4 4 3 3 2 1

1 2 2 3 3 4 3 3 2 2 1

1 1 2 2 3 3 3 2 2 1 1

1 1 2 2 3 2 2 1 1

1 1 2 2 2 1 1

1 1 2 1 1

1 1 1

1

**Data4Outfile2.txt**

PASS 1

1

1 1 1

1 1 2 1 1

1 1 2 2 2 1 1

1 1 2 2 3 2 2 1 1

1 1 2 2 3 3 3 2 2 1 1

1 2 2 3 3 4 3 3 2 2 1

1 2 3 3 4 4 4 3 3 2 1

1 2 3 4 4 5 4 4 3 2 1

1 2 3 4 5 5 5 4 3 2 1

1 2 3 4 5 6 5 4 3 2 1

1 2 3 4 5 6 5 4 3 2 1

1 2 3 4 5 5 4 3 2

1 2 3 4 5 4 3

1 2 3 4 4

1 2 3

1

------------

PASS 2

1

1 1 1

1 1 2 1 1

1 1 2 2 2 1 1

1 1 2 2 3 2 2 1 1

1 1 2 2 3 3 3 2 2 1 1

1 2 2 3 3 4 3 3 2 2 1

1 2 3 3 4 4 4 3 3 2 1

1 2 3 4 4 5 4 4 3 2 1

1 2 3 3 4 4 4 3 3 2 1

1 2 2 3 3 4 3 3 2 2 1

1 1 2 2 3 3 3 2 2 1 1

1 1 2 2 3 2 2 1 1

1 1 2 2 2 1 1

1 1 2 1 1

1 1 1

1