CSC 381 Computer Vision ( C++ )

Project 6: Distance Transform Skeleton Compression

Han Wen Loh

Soft copy: 04/01/2019

Hard copy: 04/09/2019

Algorithm Steps:

step 0: inFile 🡨 open input file

numRows, numCols, minVal, maxVal 🡨 read from inFile

dynamically allocate zeroFramedAry with extra 2 rows and 2 cols

dynamically allocate skeletonAry with extra 2 rows and 2 cols

open outFile\_1, outFile\_2

Step 1: skeletonFileName 🡨 argv[1] + “\_skeleton”

Step 2: skeletonFile 🡨 open (skeletonFileName)

Step 3: deCompressedFileName 🡨 argv[1] + “\_deCompressed”

Step 4: deCompressFile 🡨 open (deCompressedFileName)

step 5: setZero (ZeroFramedAry)

setZero (skeletonAry)

Step 6: loadImage (inFile, ZeroFramedAry) // begins at ZeroFramedAry(1,1)

Step 7: compute8Distance(ZeroFramedAry, outFile1) // Perform distance transform

Step 8: skeletonExtraction (ZeroFramedAry, skeletonAry, skeletonFile, outFile1)

// perform compression

Step 9: skeletonExpansion(ZeroFramedAry, skeletonFile, outFile2)

// perform decompression

step 10: Output numRows, numCols, newMinVal, newMaxVal to deCompressFile

Step 11: ary2File(ZeroFramedAry, deCompressFile)

// dump ZeroFramedAry to deCompressFile

Step 12: close all files

#include <iostream>

#include <fstream>

using namespace std;

class DistMedialAxis {

private:

int numRows, numCols, minVal, maxVal, newMinVal, newMaxVal;

int\*\* zeroFramedAry;

int\*\* skeletonAry;

~DistMedialAxis() {

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

delete[] zeroFramedAry[i];

delete[] skeletonAry[i];

}

delete[] zeroFramedAry;

delete[] skeletonAry;

}

void prettyPrint(int\*\* ary, ofstream& outFile) {

for (int i = 1; i <= numRows; ++i) {

for (int j = 1; j <= numCols; ++j) {

if (ary[i][j] <= 0)

outFile << " ";

else

outFile << ary[i][j] << " ";

}

outFile << endl;

}

}

int frontMin(int i, int j) {

int min = zeroFramedAry[i][j - 1];

for (int ioff = i - 1, joff = j - 1; joff < j + 2; ++joff) {

if (zeroFramedAry[ioff][joff] < min)

min = zeroFramedAry[ioff][joff];

}

return min + 1;

}

int backMin(int i, int j) {

int min = zeroFramedAry[i][j]-1 < zeroFramedAry[i][j + 1] ? zeroFramedAry[i][j] - 1 : zeroFramedAry[i][j + 1];

for (int ioff = i + 1, joff = j - 1; joff < j + 2; ++joff) {

if (zeroFramedAry[ioff][joff] < min)

min = zeroFramedAry[ioff][joff];

}

return min + 1;

}

int localMax(int i, int j) {

int max = 0;

for (int ioff = i-1; ioff < i+2; ++ioff) {

for (int joff = j-1; joff < j+2; ++joff) {

if (zeroFramedAry[ioff][joff] > max)

max = zeroFramedAry[ioff][joff];

}

}

return max;

}

void resetZero(int\*\* ary) {

for (int i = 1; i <= numRows; ++i) {

for (int j = 1; j <= numCols; ++j) {

zeroFramedAry[i][j] = 0;

}

}

}

void load(ifstream& inFile) {

int i = 0, j = 0;

inFile >> numRows >> numCols >> newMinVal >> newMaxVal;

while (!inFile.eof()) {

inFile >> i >> j;

inFile >> zeroFramedAry[i][j];

}

}

void firstPass\_8distance() {

for (int i = 1; i <= numRows; ++i) {

for (int j = 1; j <= numCols; ++j) {

if (zeroFramedAry[i][j] > 0)

zeroFramedAry[i][j] = frontMin(i, j);

}

}

}

void secondPass\_8distance() {

for (int i = numRows; i > 0; --i) {

for (int j = numCols; j > 0; --j) {

zeroFramedAry[i][j] = backMin(i, j);

}

}

}

void compute\_localMaxima() {

for (int i = 1; i <= numRows; ++i) {

for (int j = 1; j <= numCols; ++j) {

if (zeroFramedAry[i][j] > 0 && zeroFramedAry[i][j] >= localMax(i, j)) {

skeletonAry[i][j] = zeroFramedAry[i][j];

newMaxVal = zeroFramedAry[i][j];

}

}

}

}

void extraclocalMaxima(ofstream& outFile) {

outFile << numRows << " " << numCols << " " << newMinVal << " " << newMaxVal << endl;

for (int i = 1; i <= numRows; ++i) {

for (int j = 1; j <= numCols; ++j) {

if (skeletonAry[i][j] > 0) {

outFile << i << " " << j << " " << skeletonAry[i][j] << endl;

}

}

}

}

void firstPass\_Expension() {

int max = 0;

for (int i = 1; i <= numRows; ++i) {

for (int j = 1; j <= numCols; ++j) {

max = localMax(i, j);

max = max <= zeroFramedAry[i][j] ? zeroFramedAry[i][j] : max - 1;

zeroFramedAry[i][j] = max;

}

}

}

void secondPass\_Expension() {

int max = 0;

for (int i = numRows; i >= 1; --i) {

for (int j = numCols; j >= 1; --j) {

max = localMax(i, j);

max = max <= zeroFramedAry[i][j] ? zeroFramedAry[i][j] : max - 1;

zeroFramedAry[i][j] = max;

}

}

}

public:

DistMedialAxis(ifstream& inFile) {

inFile >> numRows >> numCols >> minVal >> maxVal;

newMinVal = newMaxVal = 0;

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

skeletonAry = new int\* [numCols + 2];

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

zeroFramedAry[i] = new int[numCols + 2]();

skeletonAry[i] = new int[numCols + 2]();

}

}

void loadImage(ifstream& inFile) {

for (int i = 1; i <= numRows; ++i) {

for (int j = 1; j <= numCols; ++j) {

inFile >> zeroFramedAry[i][j];

}

}

}

void compute8Distance(ofstream& outFile) {

firstPass\_8distance();

prettyPrint(zeroFramedAry, outFile);

secondPass\_8distance();

outFile << endl;

prettyPrint(zeroFramedAry, outFile);

}

void skeletonExtraction(ofstream& outFile, ofstream& skeletonFile) {

compute\_localMaxima();

outFile << endl;

prettyPrint(skeletonAry, outFile);

extraclocalMaxima(skeletonFile);

}

void skeletonExpansion(string skeletonFileName, ofstream& outFile) {

ifstream skeletonFile;

skeletonFile.open(skeletonFileName);

resetZero(zeroFramedAry);

load(skeletonFile);

skeletonFile.close();

firstPass\_Expension();

prettyPrint(zeroFramedAry, outFile);

secondPass\_Expension();

outFile << endl;

prettyPrint(zeroFramedAry, outFile);

}

void ary2File(ofstream& outFile) {

outFile << numRows << " " << numCols << " " << minVal << " " << maxVal << endl;

for (int i = 1; i <= numRows; ++i) {

for (int j = 1; j <= numCols; ++j) {

if (zeroFramedAry[i][j] >= 1)

outFile << 1 << " ";

else

outFile << 0 << " ";

}

outFile << endl;

}

}

};

int main(int args, char\*\* argv) {

string inFileName = string(argv[1]).substr(0, string(argv[1]).size() - 4);

string skeletonFileName = inFileName + "\_skeleton.txt";

string deCompressFileName = inFileName + "\_deCompressed.txt";

ifstream inFile;

ofstream outFile1, outFile2, skeletonFile, deCompressFile;

inFile.open(argv[1]);

if (inFile.fail()) {

cout << "ERROR: cannot find \"" << argv[1] << "\"\n";

exit(1);

}

outFile1.open(argv[2]);

outFile2.open(argv[3]);

skeletonFile.open(skeletonFileName);

deCompressFile.open(deCompressFileName);

DistMedialAxis\* distSkeDeComp = new DistMedialAxis(inFile);

distSkeDeComp->loadImage(inFile);

inFile.close();

distSkeDeComp->compute8Distance(outFile1);

distSkeDeComp->skeletonExtraction(outFile1, skeletonFile);

outFile1.close();

skeletonFile.close();

distSkeDeComp->skeletonExpansion(skeletonFileName, outFile2);

outFile2.close();

distSkeDeComp->ary2File(deCompressFile);

deCompressFile.close();

}

DistanceData1.txt

30 40 0 1

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 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 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 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 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 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 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 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 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 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

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 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 0 0

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

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

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

outFile1.txt

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

1 2 3 4 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 3 2 1

1 2 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 1

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

1

2

3

4

5

1 2 3 4 5 5 5 4 3 2 1

4

3

2

1

8 8

8 8

8 8

outFile2.txt

1

1 1 1

1 2 1

2 2 2 1

1 2 3 2 1

1 3 3 3 2 1

2 3 4 3 2 1

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

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

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

1 1 1 2 2 3 2 2 1 1 1

1 1 2 2 2 1 1

1 1 2 1 1

1 1 1

1

7 7 7 7 6 5 4 3 2 1

6 7 8 8 7 6 5 4 3 2 1

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

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

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

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

1 2 3 4 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 3 2 1

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

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

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

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

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

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

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

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

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

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

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

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

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

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

1 2 3 4 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 3 2 1

1 2 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 1

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

DistanceData1\_skeleton.txt

30 40 0 8

1 31 1

3 31 2

5 31 3

7 31 4

9 31 5

10 22 1

10 24 2

10 26 3

10 28 4

10 30 5

10 31 5

10 32 5

10 34 4

10 36 3

10 38 2

10 40 1

12 31 4

14 31 3

16 31 2

18 31 1

20 11 8

20 12 8

21 11 8

21 12 8

22 11 8

22 12 8

DistanceData1\_deCompressed.txt

30 40 0 1

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 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 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 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 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 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 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 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 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 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

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 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 0 0

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

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

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