****
CSCI 381-26 Project 7: Medial Axis + Distance Transform Language: JAVA

Name: Akshar Patel

Due date: Soft copy: 3/27/2020
Hard copy: 3/27/2020

Submission date: Soft copy: 3/29/2020
Hard copy: 3/29/2020

****

Part 1: Algorithm

```
*****************************
I. main (…)
*****************************
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 lossless compression


Step 9: skeletonExpansion (zeroFramedAry, skeletonFile, outFile2)
           // perform decompression

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

Step 11: ary2File (zeroFramedAry, decompressFile)

Step 12: close all files
```

Part 2: Source code

```
import java.io.*;
import java.util.*;

class Main {
    public static int numRows, numCols, minVal, maxVal;
    public static int newmin = 99999, newmax = 0;
    public static int[][] zeroFramedAry, skeletonAry;
    static void setZero(int[][] Ary){
        for(int i = 0 ; i < numRows + 2 ; i++){
```

```
        for(int j = 0 ; j < numCols + 2 ; j++){
            Ary[i][j] = 0;
        }
    }
}
static void loadImage (Scanner file, int[][] Ary){
    for(int i = 1 ; i < numRows + 1 ; i++){
        for(int j = 1 ; j < numCols + 1 ; j++){
            Ary[i][j] = file.nextInt();
        }
    }
}
static void prettyprint(int[][] Ary, PrintWriter file){
    for(int i = 1 ; i < numRows + 1 ; i++){
        for(int j = 1 ; j < numCols + 1 ; j++){
            if(Ary[i][j] == 0){
                file.print("   ");
            }
            else{
                file.print(Ary[i][j] + " ");
            }
        }
        file.println();
    }
}
static void compute8Distance(int[][] Ary, PrintWriter file){
    fistPass_8Distance(Ary);
    file.println("-----1st pass distance transform-----");
    prettyprint(Ary, file);
    secondPass8Distance(Ary);
    file.println("\n"+"-----2nd pass distance transform-----");
    prettyprint(Ary, file);
}
static void fistPass_8Distance(int[][] Ary){
    for(int i = 1 ; i < numRows + 1 ; i++){
        for(int j = 1 ; j < numCols + 1 ; j++){
            if(Ary[i][j] > 0){
                int a = Ary[i-1][j-1], b = Ary[i-1][j], c = Ary[i-
                       1][j+1], d = Ary[i][j-1];
                Ary[i][j] = Math.min(Math.min(a, b), Math.min(c, d)) + 1;
            }
        }
    }
}
static void secondPass8Distance(int[][] Ary){
    for(int i = numRows ; i > 0 ; i--){
        for(int j = numCols ; j > 0 ; j--){
            if(Ary[i][j] > 0){
                int e = Ary[i][j+1], f = Ary[i+1][j-
                  1], g = Ary[i+1][j], h = Ary[i+1][j+1], x = Ary[i][j];
                Ary[i][j] = Math.min(x, 1+Math.min(Math.min(e, f), Math.min(g, h)));
            }
            if(newmin > Ary[i][j]){
                newmin = Ary[i][j];
            }
            if(newmax < Ary[i][j]){
                newmax = Ary[i][j];
```

```
            }
        }
    }
}
static void skeletonExtraction(int[][] Ary, int[][] skAry, PrintWriter skfile,
 PrintWriter file){
    computeLocalMaxima(Ary, skAry);
    file.println("\n"+"-----Local maxima-----");
    prettyprint(skAry, file);
    extractLocalMaxima(skAry, skfile);
    skfile.close();
}
static void computeLocalMaxima(int[][] Ary, int[][] skAry){
    for(int i = 1 ; i < numRows + 1 ; i++){
        for(int j = 1 ; j < numCols + 1 ; j++){
            if(Ary[i][j] > 0 && isLocalMaxima(Ary, i, j) != 0){
                skAry[i][j] = Ary[i][j];
            }
            else{
                 skAry[i][j] = 0;
            }
        }
    }
}
static int isLocalMaxima(int[][] Ary, int i, int j){
    int a = Ary[i-1][j-1], b = Ary[i-1][j], c = Ary[i-1][j+1], d = Ary[i][j-1];
    int e = Ary[i][j+1], f = Ary[i+1][j-1], g = Ary[i+1][j], h = Ary[i+1][j+1];
    int x = Ary[i][j];
    if(x >= a && x >= b && x >= c && x >= d && x >= e && x >= f && x >= g && x >= h){
        return 1;
    }
    else{
        return 0;
    }
}
static void extractLocalMaxima(int[][] skAry, PrintWriter skfile){
    skfile.println(numRows + " " + numCols + " " +  newmin + " " + newmax);
    for(int i = 1 ; i < numRows + 1 ; i++){
        for(int j = 1 ; j < numCols + 1 ; j++){
            if(skAry[i][j] > 0){
                skfile.println(i + " " + j + " " + skAry[i][j]);
            }
        }
    }
}
static void skeletonExpansion(int[][] Ary, Scanner skfile, PrintWriter file){
    setZero(Ary);
    load(skfile, Ary);
    firstPassExpension(Ary);
    file.println("-----1st pass Expansion-----");
    prettyprint(Ary, file);
    secondPassExpension(Ary);
    file.println("\n"+"-----2nd pass Expansion-----");
    prettyprint(Ary, file);
}
static void load(Scanner file, int[][] Ary){
    file.nextLine();
```

```
    while(file.hasNextInt()){
        Ary[file.nextInt()][file.nextInt()] = file.nextInt();
    }
}
static void firstPassExpension(int[][] Ary){
    for(int i = 1 ; i < numRows + 1 ; i++){
        for(int j = 1 ; j < numCols + 1 ; j++){
            if(Ary[i][j] == 0){
                int x = max_neighbors(i, j, Ary) - 1;
                if(Ary[i][j] < x){
                    Ary[i][j] = x;
                }
            }
        }
    }
}
static void secondPassExpension(int[][] Ary){
    for(int i = numRows ; i > 0 ; i--){
        for(int j = numCols ; j > 0 ; j--){
            int x = max_neighbors(i, j, Ary);
            if(Ary[i][j] < x){
                Ary[i][j] = x - 1;
            }
        }
    }
}
static int max_neighbors(int i, int j, int[][] Ary){
    int max = 0;
    for(int x = i - 1 ; x < i + 2 ; x++){
        for(int y = j - 1 ; y < j + 2 ; y++){
            if (Ary[x][y] > max) {
                max = Ary[x][y];
            }
        }
    }
    return max;
}
static void ary2File(int[][] Ary, PrintWriter file){
    file.println(numRows+" "+numCols+" "+minVal+" "+maxVal);
    for(int i = 1 ; i < numRows + 1 ; i++){
        for(int j = 1 ; j < numCols + 1 ; j++){
            if(Ary[i][j] >=1 ){
                file.print(1 + " ");
            }
            else{
                file.print(0 + " ");
            }
        }
        file.println();
    }
}
public static void main(String[] args) throws IOException {
    Scanner inFile = new Scanner(new FileInputStream(args[0]));
    PrintWriter outFile1 = new PrintWriter(new FileOutputStream(args[1]));
    PrintWriter outFile2 = new PrintWriter(new FileOutputStream(args[2]));
    numRows = inFile.nextInt();
    numCols = inFile.nextInt();
```

```
        minVal = inFile.nextInt();
        maxVal = inFile.nextInt();
        zeroFramedAry = new int[numRows + 2][numCols + 2];
        skeletonAry = new int[numRows + 2][numCols + 2];
        String skeletonFileName = args[0];
        PrintWriter skeletonFile = new PrintWriter(skeletonFileName.substring(0,
           skeletonFileName.indexOf(".")) + "_skeleton"+ ".txt");
        String decompressedFileName = args[0];
        PrintWriter decompressFile  = new PrintWriter(decompressedFileName.substring(0,
           decompressedFileName.indexOf(".")) +"_decompressed"+ ".txt");
        setZero(zeroFramedAry);
        setZero(skeletonAry);
        loadImage(inFile, zeroFramedAry);
        compute8Distance(zeroFramedAry, outFile1);
        skeletonExtraction(zeroFramedAry, skeletonAry, skeletonFile, outFile1);
        File skefile = new File(skeletonFileName.substring(0, skeletonFileName.indexOf
           (".")) + "_skeleton"+ ".txt");
        Scanner skeletonFile_2 = new Scanner(skefile);
        skeletonExpansion(zeroFramedAry, skeletonFile_2, outFile2);
        ary2File(zeroFramedAry, decompressFile);
        inFile.close();
        outFile1.close();
        outFile2.close();
        skeletonFile_2.close();
        decompressFile.close();
    }
}
```

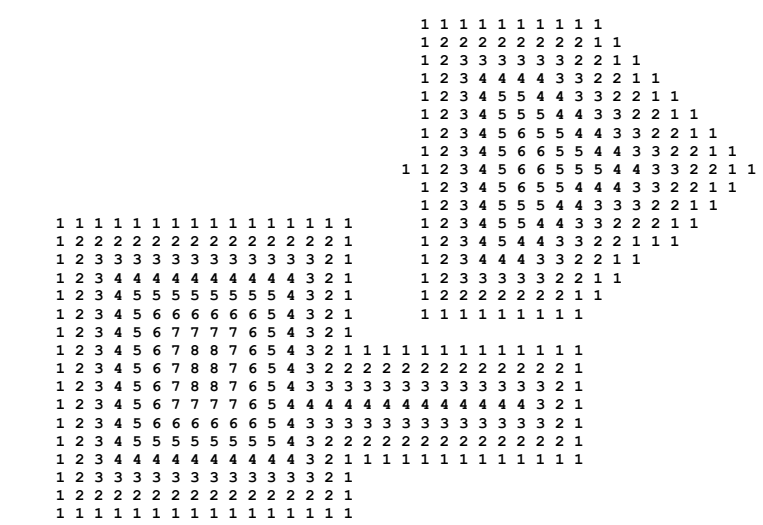
Part 3: Output

- **image1**

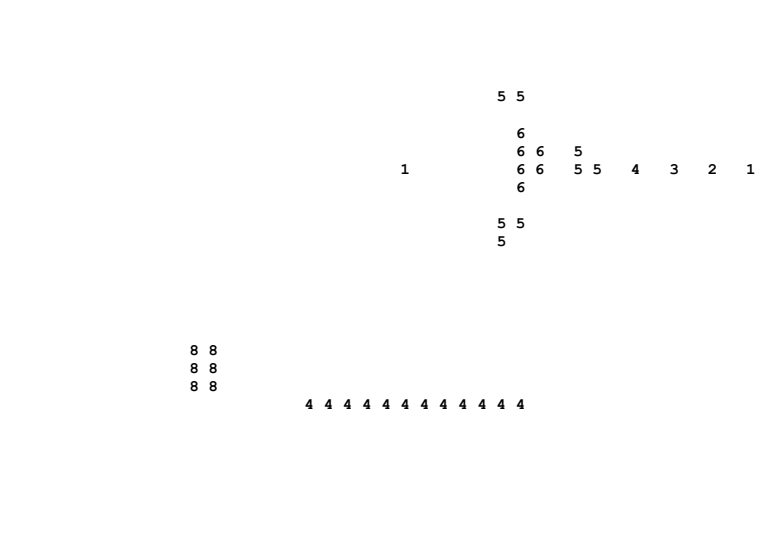
<table>
<tr><th>input file</th><th>outFile1</th></tr>
<tr><td><pre>
30 40 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 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 0 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 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 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 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 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 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 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 1 1 1 0 0 0 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 1 1 1 1 1 0 0 0 1 1 1 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 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 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 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 1 1 1 1 1 1 1 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 1 1 1 1 1 1 1 1 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 1 1 1 1 1 1 1 1 1 1 1 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 1 1 1 1 1 1 1 1 1 1 1 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 1 1 1 1 1 1 1 1 1 1 1 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 1 1 1 1 1 1 1 1 1 1 1 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 1 1 1 1 1 1 1 1 1 1 1 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 1 1 1 1 1 1 1 1 1 1 1 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 1 1 1 1 1 1 1 1 1 1 1 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 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
</pre></td><td><pre>
-----1st pass distance transform-----

                                            1 1 1 1 1 1 1 1 1 1
                                            1 2 2 2 2 2 2 2 2 1 1
                                            1 2 3 3 3 3 3 3 2 2 1 1
                                            1 2 3 4 4 4 4 3 3 2 2 1 1
                                            1 2 3 4 5 5 4 4 3 3 2 2 1 1
                                            1 2 3 4 5 5 5 4 4 3 3 2 2 1 1
                                            1 2 3 4 5 6 5 5 4 4 3 3 2 2 1 1
                                            1 2 3 4 5 6 6 5 5 4 4 3 3 2 2 1 1
                                          1 1 2 3 4 5 6 6 6 5 5 4 4 3 3 2 2 1 1
                                            1 2 3 4 5 6 7 6 6 5 5 4 4 3 3 2 2
                                            1 2 3 4 5 6 7 7 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 7 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 8 7 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 7 8 8 7 7 6
      1 2 3 4 4 4 4 4 4 4 4 4 4 3 2 1   1 2 3 4 5 6 7 8 8 8 7
      1 2 3 4 5 5 5 5 5 5 5 5 4 3 2 1   1 2 3 4 5 6 7 8 9 8
      1 2 3 4 5 6 6 6 6 6 6 5 4 3 2 1   1 2 3 4 5 6 7 8 9
      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 1 1 1 1 1 1 1 1 1 1 1
      1 2 3 4 5 6 7 8 8 7 6 5 4 3 2 2 2 2 2 2 2 2 2 2 2 2 2 1
      1 2 3 4 5 6 7 8 8 7 6 5 4 3 3 3 3 3 3 3 3 3 3 3 3 3 2 1
      1 2 3 4 5 6 7 8 8 7 6 5 4 4 4 4 4 4 4 4 4 4 4 4 4 3 2 1
      1 2 3 4 5 6 7 8 8 7 6 5 5 5 5 5 5 5 5 5 5 5 5 5 4 3 2 1
      1 2 3 4 5 6 7 8 8 7 6 6 6 6 6 6 6 6 6 6 6 6 6 5 4 3 2 1
      1 2 3 4 5 6 7 8 8 7 7 7 7 7 7 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 8 8 8 8
      1 2 3 4 5 6 7 8 9 9 9 9 9 9 9 1
      1 2 3 4 5 6 7 8 9 10 10 10 10 10 2 1
</pre></td></tr>
</table>

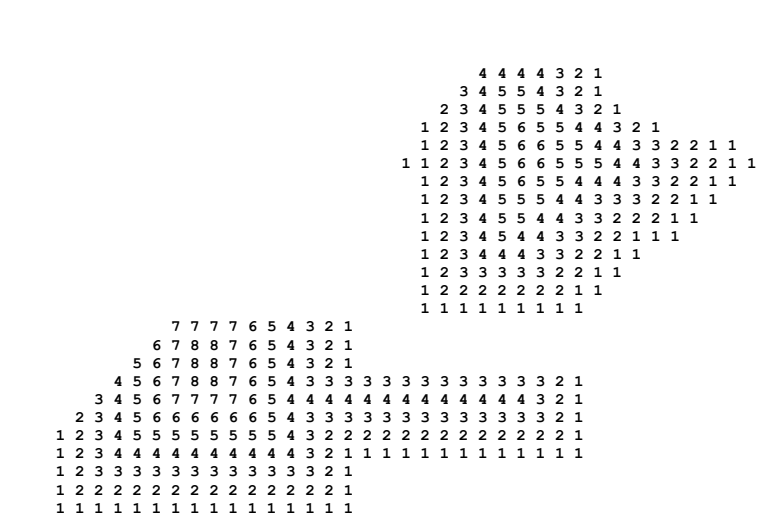
outFile1



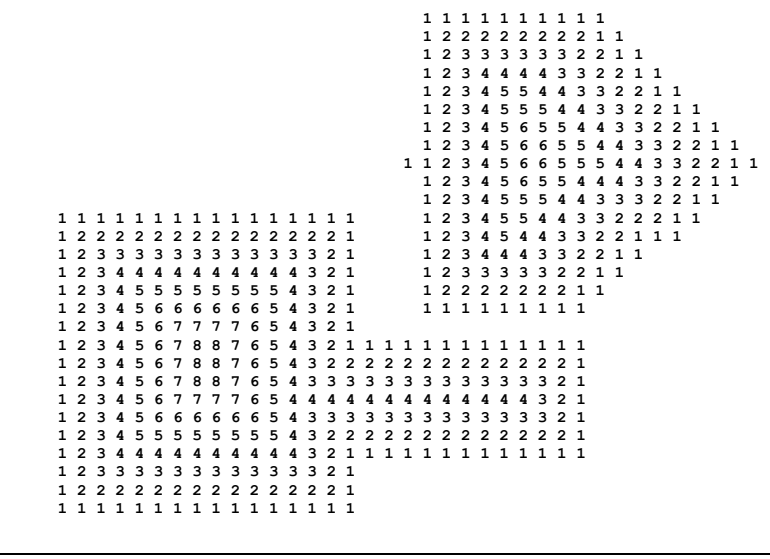
outFile1



outFile2



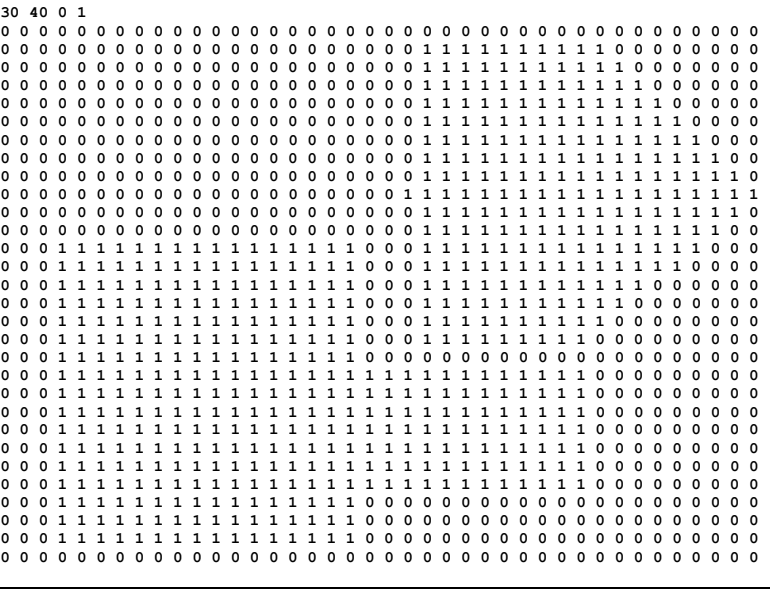
outFile2



skeleton file

```
30 40 0 8
6 27 5
6 28 5
8 28 6
9 28 6
9 29 6
9 31 5
10 22 1
10 28 6
10 29 6
10 31 5
10 32 5
10 34 4
10 36 3
10 38 2
10 40 1
11 28 6
13 27 5
13 28 5
14 27 5
20 11 8
20 12 8
21 11 8
21 12 8
22 11 8
22 12 8
23 17 4
23 18 4
23 19 4
23 20 4
23 21 4
23 22 4
23 23 4
23 24 4
23 25 4
23 26 4
23 27 4
23 28 4
```

decompressed file



- **image2**

```
49 64 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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 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

```
-----1st pass distance transform-----

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

outFile1

```
-----2nd pass distance transform-----

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

outFile1

```
-----Local maxima-----

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

outFile2

```
-----1st pass Expansion-----

                                                  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 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 3 2 2 1 1
                                          1 1 2 2 3 3 2 2 1 1
                                            1 1 2 2 2 2 1 1
                                              1 1 2 2 2 1 1
                                                1 1 2 1 1
                                                  1 1 1
                                                    1
                                                    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 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 3 2 2 1 1
                                          1 1 2 2 3 3 2 2 1 1
                                            1 1 2 2 2 2 1 1
                                              1 1 2 1 1
                                                1 1 1
                                                  1
```

outFile2

```
-----2nd pass Expansion-----

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

skeleton file

```
49 64 0 5
4 31 1
6 31 2
8 31 3
10 31 4
12 31 5
13 22 1
13 24 2
13 26 3
13 28 4
13 30 5
13 31 5
13 32 5
13 34 4
13 36 3
13 38 2
13 40 1
14 31 5
16 31 4
18 31 3
20 8 5
20 9 5
20 10 5
20 11 5
20 12 5
20 13 5
20 14 5
20 15 5
20 31 2
21 8 5
21 9 5
21 10 5
21 11 5
21 12 5
21 13 5
21 14 5
21 15 5
22 31 1
23 31 1
25 31 2
27 31 3
29 31 4
31 31 5
32 22 1
```

```
32 24 2
32 26 3
32 28 4
32 30 5
32 31 5
32 32 5
32 34 4
32 36 3
32 38 2
32 40 1
33 31 5
35 31 4
37 31 3
39 31 2
41 31 1
```

- **decompressed file**

```
49 64 0 1
```