1	2	3	4	5	6	7	8	9	=
13	11	14	10	13	13	13	15	6	112

CV Spring 2021 Exam I (Part 1 - Essay questions) Name: Andrew Alleyne

***** Reminding you that this exam is a closed book exam. Rules are given in the online exam instruction posted in google classroom and if you cheat part of part of this exam, -999 will be the score of this exam.**

1. What is the run-length encoding for? (1)
2.

Run length encoding is for noise compression.

3. What is a run in the run-length encoding? (1)
   The run in run length encoding is deals with scanning the image and recording the adjacent consecutive pixels of the same value.

4. How do we represent a run in the run-length encoding? (1)

   We can represent a run length encoding by as such: startRow, StartCol, Grayscale level and pixelCount. For instance, 1,1,7,400 would represent start row is 1, start col is 1, grayscale value is 7 and the number of pixel count of grayscale values are 400.

5. Among the four run-length encoding methods, which method has the best compression rate? And which method has the worst compression rate? (2)

   Among the 4 of the run-length encoding methods:
   a. Encoding 0's and no wrap around is the worst.
   b. No 0's and no wrap around is the best.

6. Among the four run-length encoding methods, which method is used in fax machine? (1)

   The run-length method used in fax machines is Encode 0's , wrap around.

7. What kind of image would yield the worst compression rate to use run-length encoding no matter which method to use? Illustrate an example. (2)

   An image where each pixel value is a different number yields the worst compression rate. We would have to assign 4 bytes for each number which increases the compression size 4 times.

8. Write the results of the run-length encoding using: (A) Include zero's and wrapped around and (B) no zero's and wrapped around. Do not forget to write the image header! (5)

7 10 0 9 ←image header

	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	3	3	3	3
1	3	3	3	3	3	3	3	3	3	3
2	3	3	4	4	4	4	4	4	4	4
3	7	7	7	7	7	7	5	5	5	5
4	5	5	5	1	1	1	1	0	0	0
5	6	6	6	6	6	6	6	6	0	0
6	0	0	0	0	0	0	0	0	0	0

(A)

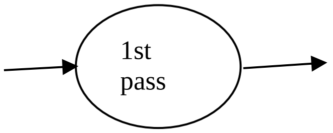
0	0	0	6
0	6	3	16
2	2	4	8
3	0	7	6
3	6	5	7
4	3	1	4
4	7	0	3
5	0	6	8
6	0	0	12

(B)

0	6	3	16
2	2	4	8
3	0	7	6
3	6	5	7
4	3	1	4
5	0	6	8

9. Given the image on the left, apply the **<u>1st pass</u>** of the **<u>8-connected</u>** component algorithm, modify the original image on the right to show the result. You must initialize the EqTable and update it during the process, and as will as update newLabel. (6)

0	0	0	0	0	0	0	0	0	0
0	1	0	0	1	0	0	1	0	0
0	0	1	0	0	1	0	1	0	0
0	0	0	1	0	1	0	0	1	0
0	1	1	0	1	1	0	0	1	0
0	1	0	0	0	0	0	1	0	0
0	0	0	1	0	1	0	0	1	0
0	0	0	1	0	0	1	0	1	0
0	1	1	0	0	0	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	0	0	2	0	0	3	0	0
0	0	1	0	0	2	0	3	0	0
0	0	0	1	0	2	0	0	3	0
0	1	1	0	2	2	0	0	3	0
0	1	0	0	0	0	0	3	0	0
0	0	0	1	0	1	0	0	3	0
0	0	0	1	0	0	1	0	3	0
0	1	1	0	0	0	1	3	0	0
0	0	0	0	0	0	0	0	0	0

NewLabel: 0

EqTable	0	1	2	3	4	5	6	7	8	9	10	11	12	:
	0	1	2	3										:

10. Apply the **<u>2st pass</u>** of the **<u>8-connected</u>** component algorithm on the left (the result of the 1st pass 8-CC), and modify the original given on the right to show the result. Update EqTable during the process as needed. (5)

$$\begin{array}{cccccccccc}
0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&2&0&3&0\\
0&4&0&0&1&0&0&2&0&0\\
0&0&4&1&0&1&0&0&2&0\\
0&0&1&0&0&1&0&2&0&0\\
0&0&0&0&0&0&0&2&0&0\\
0&5&5&0&0&6&0&0&2&0\\
0&0&0&5&0&0&6&0&2&0\\
0&7&0&0&0&0&6&2&0&0\\
0&0&0&0&0&0&0&0&0&0
\end{array}
\xrightarrow{\text{2nd pass 8 -}}
\begin{array}{cccccccccc}
0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&2&0&2&0\\
0&1&0&0&1&0&0&2&0&0\\
0&0&1&1&0&1&0&0&2&0\\
0&0&1&0&0&1&0&2&0&0\\
0&0&0&0&0&0&0&2&0&0\\
0&5&5&0&0&2&0&0&2&0\\
0&0&0&5&0&0&2&0&2&0\\
0&7&0&0&0&0&2&2&0&0\\
0&0&0&0&0&0&0&0&0&0
\end{array}$$

NewLabel : 7

EqTable	0	1	2	3	4	5	6	7	8	9	10	11	12	. .
	0	1	2	2	1	2	2	7	8	9	10	11	12	. .

11. Given the 3x3 diagram below, describe in cases, how to assign pixel x's label, during the1st pass and the 2nd pass of the **4-**connected component algorithm.

a	b	c
d	x	e
f	g	h

a) 1st pass of **4-**connected component: (3)

Case 1: a = b = c = d = 0
If this is true, we create a new label for x. **NewLabel** ++ where x =1
The current pixels value will be incremented by 1.
We then must make sure we Update equivalency table.

Case 2: If some/all the neighbors are the same we have to write the same label.
Case 3: some/all the neighbors already have labels but are different then we take the minimum.
Min of neighboring labels → min (x, a, b, c, d)

Update equivalence table.

b) 2nd pass of **4-**connected component: (3)

The 2^{nd} pass does not vary much. We do not assign new labels here.

Case 1: e = f = g = h = 0 Then the current pixel retains its label.
Case 2: e = f = g = h = current Pixel then the current pixel retains its label.
Case 3: At least 2 amongst the e, f, g, h and the currentPixel have different labels excluding 0 (background) .
minLabel <--- min(of all neighbors e, f, g, h and the currentPixel)
If the current pixel is > than the minimum label
We update equivalence table of the current label to the minLabel. EQTable [currentPixel];

```
X  <--- minLabel
```

12. What is morphology? (1)

    Morphology is study of shapes.

13. Given a 1-D image and the structuring element (with origin **1**) below, apply 1-D morphological “dilation”; write the result after arrow →. (2)

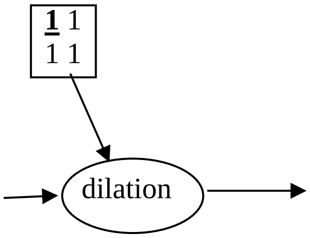
```
                    0 0 1 1 0 1 0 1 1 0 1 0 0 1 0 1 0 0        1 1 1
Dilation result →   0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0
```

14. Given the 1-D image and the structuring element (with origin **1**) below, apply 1-D morphological “erosion” and write the result after arrow →. (2)

```
                    0 0 1 1 1 1 0 1 1 1 1 0 0 1 0 1 0 0        1 1 1
Erosion result →    0 0 0 1 1 0 0 0 1 1 0 0 0 0 0 0 0 0
```

15. Given a 2-D image on the left and a structuring element (with origin **1**) on the top, apply 2-D ‘**dilation’** operation using the structuring element; modify the image given on the right to show the result of dilation. (3)

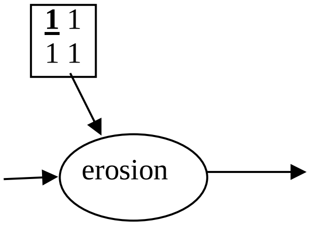
```
0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 1 0
0 0 0 0 1 0 0 0 1 0
0 1 0 1 0 0 1 0 0 0
0 0 1 0 0 1 0 0 1 0
0 0 0 0 0 0 0 0 0 0
```



```
0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 1 0 0 1 1
0 0 0 0 1 1 0 0 1 1
0 1 1 1 1 0 1 1 0 0
0 1 1 1 1 1 1 1 1 1
0 0 1 1 0 1 1 0 1 1
```

16. Given a 2-D image on the left and a structuring element (with origin **1**) on the top, apply 2-D ‘**erosion’** operation using the structuring element; modify the image given on the right to show the result of erosion. (3)

```
0 0 0 0 0 0 0 0 0 0
0 1 0 0 1 0 1 0 1 0
0 0 1 1 0 1 1 0 0 0
0 1 1 1 0 1 1 0 0 0
0 1 1 1 0 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 0 0 1 0 0 0 0
0 1 1 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
```

17. In general, when do we use the Opening operation? Explain. (1)

    We use the opening operation when we want to extract and object from the image.

18. In general, when do we use the Closing operation? Explain. (1)

    We use the closing operation when we want to fill the background. Closing can also be used to fill in the salts/noise.

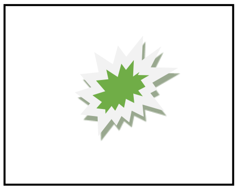
19. Given a binary image, where objects (1's) in the image are spaghetti, meat balls, small pieces of meat. The task is to extract only meat balls, no spaghetti, and no small pieces of meat. Design a sequence of morphological operations with the shape, the size, and the origin of the structuring element for the task. You may draw an object-process diagram or write in English showing the steps by steps of the sequence of morphological operations and the expected result after each operation. (5)

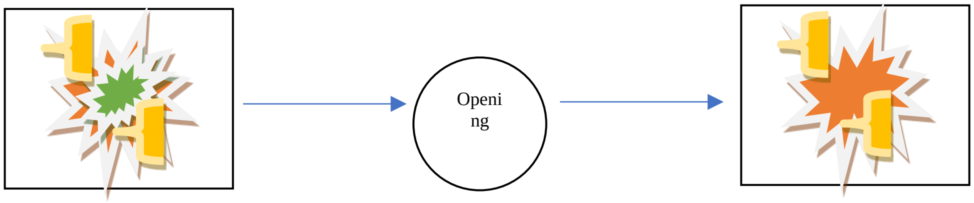
We can design a structuring element that is 0.5 bigger than the small pieces of meat and 0.5 smaller than the meatball.

However, supposing the meatball is covered with small pieces of meat and spaghetti. So, we design structuring element to first remove the pieces of meat and then the spaghetti.

Pieces of meat will be 0.5 smaller than the pieces of meat but larger than the size of the spaghetti.
Its origin: center

Structuring element for the pieces of meat →



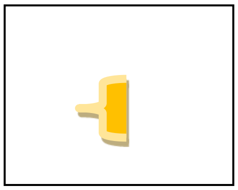


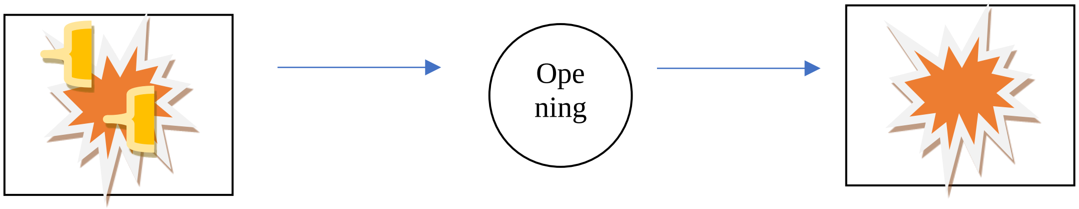
= Will extract pieces of meat. Will be left with spaghetti and meatballs.

Opening

Structuring element for the meatball
spaghetti dimensio
Origin: center

Structuring element of the spaghetti will be smaller than the meatball and 0.5 smaller than the meatball →





We are left with the image of the meatball.

20. What is the main purpose of framing the input image prior to image processing operations? (2)

We frame the input image prior to image processing so that we can operate on it without going out of bounds.

21. Given a 3x3 neighbors of pixel, p (i, j) below, write the computation formula for the three image noise filters you were taught in class: (4)

a	b	c
d	(i,j)	e
f	g	h

(a) 3x3 averaging: p’(i,j) =

(b) 3x3 median-filter: p’(i,j) =

Mask

1	2	1
2	4	2
1	2	1

(c) Gaussian filter (with a given 3x3 Mask):

p’(i,j) =

22. Why the 3x3 median-filter and the 3x3 averaging-filter may wipe out corners of objects in the image? Explain your answer for credits. (2)

The 3x3 median filter may wipe out corners because of the pixel choice.

23. Write the algorithm steps for the corner-preserve-averaging filter. (5)

24. Why do we need the image header in every image? (2)

    The image header provide information such as the row, col , min grayscale values and max grayscale values.

25. What is the histogram of an image? (1)

The histogram of an image quantifies the intensity of the number of pixels in an image. In it is stored in the hist[i] array and the relative index is incremented by 1 each time it encounters a pixel pertaining to that index.

26. Can we reconstruct the image from its histogram? Explain your answer. (1)

    We cannot reconstruct an image from a histogram because a histograms purpose is to tell us the intensity of pixels.

27. Can we guess the content of an image from its histogram? Explain your answer. (1)

    It is possible to guess the contents of an image from a histogram as it just provides pixel intensity information. However we can guess if the image has dark or lighter object in it .

28. Will the histogram of a grey-scale image changed drastically after a noise cleaning operation from before the operation? Explain your answer for credits. (2)

    Yes, the histogram of a greyscale image will change immensely after using a noise cleaning operation prior to obtaining the histogram for the given image. Image enhancement aids in nose suppression by using either mean, median, corner and 2d gaussian filtering. These algorithms will change the pixel values thus resulting in a difference in the histogram.

29. Write in English or in algorithm steps for the general idea of the deepest**-concavity** automatic threshold selection method. (4)

- Given some image we probe it left to right and top to bottom.
- We first must smooth the image out using the median filter, which will be able to store 5 pixels.
- We do this by placing the pixels at the current pixel into the median array and sorting them.
- Then select the median/middle and place into a new array, A'.
- We continue scanning each pixel until the entire image is processed.
- When the image is done processing, we then find the peaks of the now smoothed graph.
- When we've obtained those peak, we then find the slope of them.
- We then want to find the difference between the slope and each value in between the peaks.
- We then return the final max of all of the differences as the threshold.

30. Write in English or in algorithm steps for the general idea of the **bi-means** automatic threshold value selection. (4)

CV Spring 2021 Exam I (Part 2 - coding) Name:___________________

*** **When ask to write a method, you MUST write the header/prototype of the method, NOT just method body.**

31. (In C++) Implement the following steps (write in the blank space after steps):

Step 0: Write C++ library you include in your C++ projects. (2)

```
#include <fstream>
```

Step 1: inFile, outFile ← Open the input and output files from argv (1st and 2nd), (2)
thrVal ← get from 3rd argv. (1)

```
Ifstream inFile;
inFile.open(argv[1]);

ofstream outFile;
outFile.open(argv[2]);
```

```
int threshold;
threshold = argv[3];
```

```
Step 2: data ← inFile
             If data <  thrVal
                  outFile ← 0
            else
                  outFile ← data
```

```
int data;
inFile >> data;

if( data < thrVal){
      outFile << 0 ;
      }
else {
     outFile << data ;
}
```

Step 3: repeat step 2 until at the end of inFile. (4)

```
while (inFile >> data){

      if( data < thrVal){
                  outFile << 0 ;
                  }
      else {
            outFile << data ;
      }
}
```

32. **<u>(In C++)</u>** a) Dynamically allocates the 2D mirrowFramedAry (use MFAry for short) size of numRows + frameSize by numCols + frameSize. (b) Reads and loads data from inFile into inside frame of MFAry. (5)

```
                  Int numRows ;
```

```
        int numCols;
        int frameSize;

        int MF [][];
        MF =new mf[numRows + frameSize];
        For(int i=0; i < numCols + frameSize;  i++) {
                MF = * mf[i];

        }
```

33. **(in C++)** Write the method, mirrorFraming (MFAry, frameSize) for 3x3 median filter where frameSize is 3. (5)

```
 For(int I = 0; I < numRows; i++){
        For(int k = 0; k < numCols; k++){


        }
}
```

34. **(in C++)** Write the method, convolution (MFAry, i, j, mask), the method computes the convolution using a given 5x5 mask onto the pixel MFAry[i, j]'s 5x5 neighborhood and returns the convolution result. (6)

35. **(In Java)** Implement the following steps. (4)

```
step 1: i ← rowFrameSize
step 2: j ← colFrameSize
step 3: if inAry [i,j] > 0
```

```
                    dilation (i, j, inAry, outAry)  // call dilation method without define it.  You will write the dilation method next.
step 4: j++
step 5: repeat step 3 to step 4 while j <  (numImgCols + colFrameSize)
step 6: i++
step 7: repeat step 2 to step 6 while i <  (numImgRows + rowFrameSize)
```

36. **(In Java)** Write the method, dilation (i, j, inAry, outAry, structAry) that performs dilation on inAry [i, j] with structAry and outputs to outAry[i, j]. You may use rowOrigin, colOrigin, numStructRows, and numStructCols without define them. (6)

37. **This question is for graduatae students who registered in cs 780 . Undergrade will get extra points if answered.** **(In Java)** Write the method, erosion (i, j, inAry, outAry, structAry) that performs eroson on inAry [i, j] with structAry and outputs to outAry[i, j]. You may use rowOrigin, colOrigin, numStructRows, and numStructCols without define them. (-4 to 0). If you get it wrong, -4; if correct, 0, partial correct from -1 to -3, depends on the correctness.

```
Void erosion( int I, int j, int[][] inAry, outAry[][], structAry[][]){

Boolean hasChecked = false;

For(int k = 0; k < numStructRows; k++){

        For(int m = 0; m < numStructCols; m++){

                If(inAry [ I + k + numstructRows] [ I + m + numstructRows]!= numStructRows && structAry[k][m] !=1)

                outAry[ I + k + numstructRows] [ I + m + numstructRows] =0


                Boolean hasChecked = false;

                }

                If(hasChecked == false) Break;



        } //inner loop

                If(!hasChecked == true) Break;

                If(hasChecked == true){
                outAry[ I + k + numstructRows] [ I + m + numstructRows] = 1;

                }



}
```