CGIP-Module-5-Important-Topics

1. Contrast stretching

What is contrast stretching

- Contrast stretching is an Image enhancement method which attempts to improve an image by stretching the range of intensity values
- We will stretch the minimum and maximum intensity values present to the possible minimum and maximum intensity values
    - For example: If minimum intensity value rmin = 100
        - Then its stretched to possible intensity value 0
    - If max intensity value rmax is less than possible maximum intensity value 255, then its stretched out to 255
- 0-255 is taken as standard minimum and maximum intensity values for 8 bit images

General Formula for Contrast Stretching:

$$s = (r - r_{min}) \frac{(I_{max} - I_{min})}{(r_{max} - r_{min})} + I_{min}$$

For I min = 0 and I max = 255 (for standard 8-bit grayscale image)

$$s = 255 \times \frac{(r - r_{min})}{(r_{max} - r_{min})}$$

where,

r = current pixel intensity value

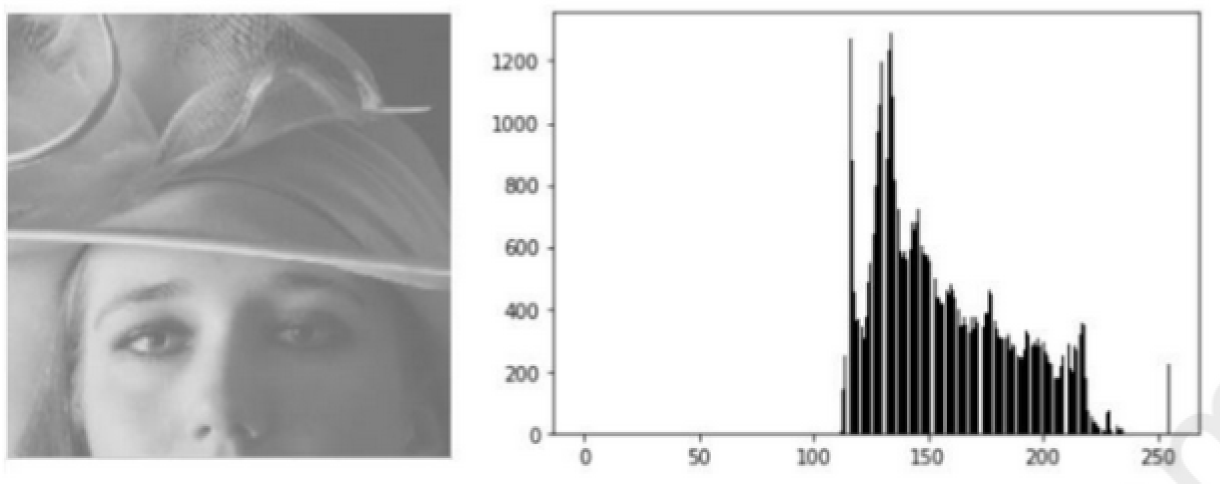
r min = minimum intensity value present in the whole image

r max = maximum intensity value present in the whole image

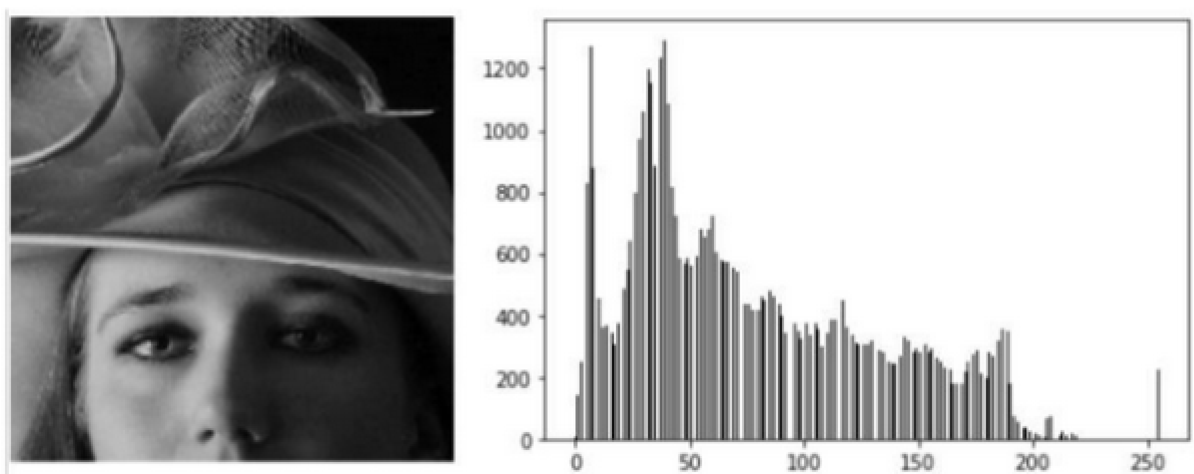
Note: Output intensity value s should be rounded up to nearest integer value.

A brief idea of how Contrast stretching looks like

Before



After



2. Histogram equalization problem

What is a histogram?

- Histogram is a graphical representation of intensity distribution of an image
- Its basically the number of pixels for each intensity value considered

Histogram equalization

- Histogram equalization is a computer image processing technique used to improve contrast in images
- It accomplishes this by effectively spreading out the most frequent intensity values

- Basically stretching out the intensity range of the image

What is the use of histogram equalization

- In digital image processing, the contrast of image is enhanced using this technique
- Its used to increase the spread of the histogram

Algorithm

1. Find the frequency of each value represented on the horizontal axis of the histogram
    1. Here each value is the intensity in case of image
2. Calculate the probability density function for each intensity value
3. After finding the PDF, calculate the cumulative density function for each intensities frequency
4. The CDF value is in the range 0-1, so we multiply all CDF values by the largest value of intensity
5. Round off the final values to integer values

Example Problem-1

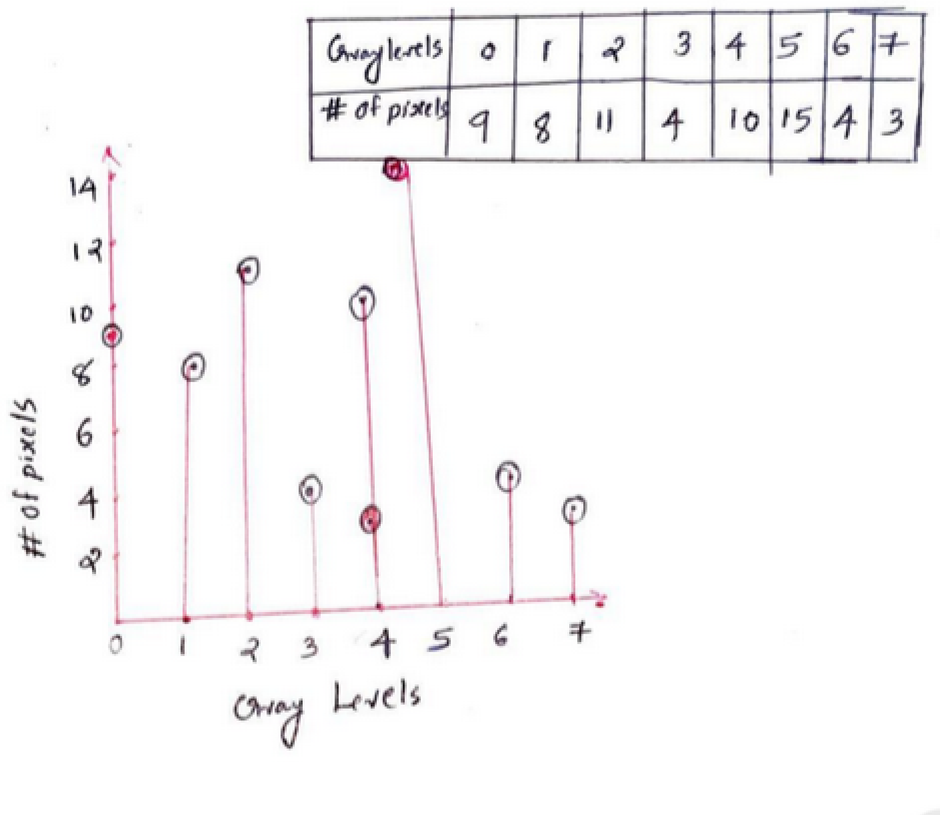
Perform the histogram equalization for an 8*8 image shown below.

Gray levels	0	1	2	3	4	5	6	7
No. of pixels	9	8	11	4	10	15	4	3

First we need to plot the histogram for the input page

- Here the X axis are the gray level

- The Y axis is the number of pixels



Next we need to fill the table

- Given the Image is 8 x 8 = 64
  - n = 64
- Gray levels = rk
- No of pixels = nk
- Probability Density function(PDF) = No of pixels / n
  - Here n is 64
- Cumulative Density function (CDF), We add the PDFS one by one

Gray levels(rk)	No. of pixels (nk)	Probability density function(PDF) nk/n	Cumulative density function(CDF) Sk	Multiply with largest intensity(7) Sk*7	Histogram equalized levels (round off)
0	9	9/64	9/64	0.9843	1
1	8	8/64	17/64	1.859	2
2	11	11/64	28/64	3.062	3
3	4	4/64	32/64	3.5	4
4	10	10/64	42/64	4.593	5
5	15	15/64	57/64	6.234	6
6	4	4/64	61/64	6.671	7
7	3	3/64	1	7	7

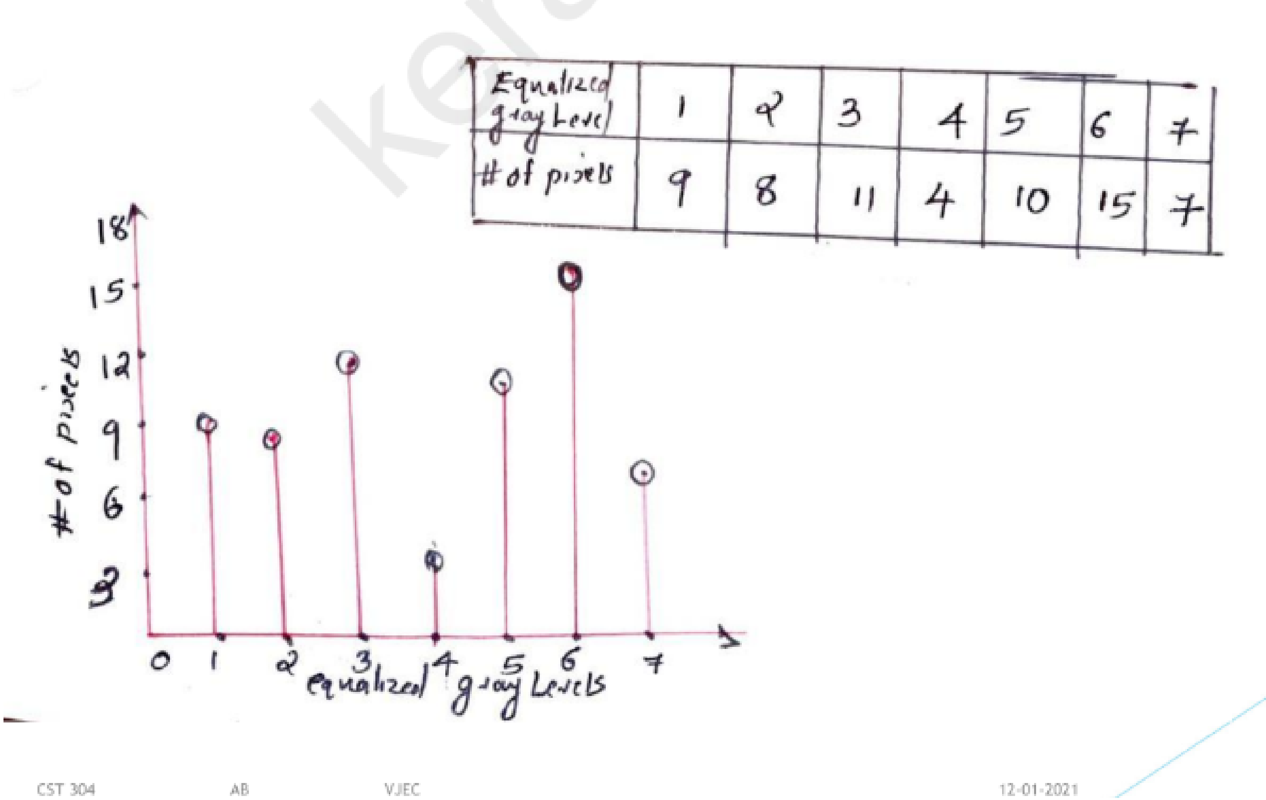
n=64

- Here 6 and 7 has the same Histogram equalized levels (7)
- So they can be combined together

Plot histogram for the equalized image

Equalized gray levels	1	2	3	4	5	6	7
No.of pixels	9	8	11	4	10	15	7 (4+3)

- Based on the above table make a new histogram



3. Problems from different filters(Median, mean,min, max)

Median filter

- Replace the value of a pixel by the median of the gray level in the neighbourhood of that pixel
- Application
  - Most popular filter
  - Excellent noise reduction
  - Less blurring

Problem

Apply median filters for the given image

10	20	20
20	15	20
20	25	100

-
- Rank pixels in the increasing value of intesnity
- 10,15,20,20,20,20,20,25,100
- Find the median value
  - 10,15,20,20,**20**,20,20,25,100
  - 20 is median

Max Filters

- Its used for finding brightest point in the image

Apply max filters for the given image

10	20	20
20	15	20
20	25	100

- Brightest point is 100

Min Filters

- Its used for finding the brightest point in the image
- Apply min filters

10	20	20
20	15	20
20	25	100

-
- Darkest point is 10

Box Filter / Mean / Average Filter

- Its used for blurring and noise reduction

- Blurring - removal of small details from an image, prior object extraction(Preprocessing)
- Noise reduction can be implemented by blurring with linear/non linear filter.

Problem-1

A 5x5 image patch is shown below. Compute the value of the pixel at the position (3x3) if it is smoothened by a 3x3 average filter.

Pixel value at the position (3x3) = 5

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 2 \\ 5 & 6 & 7 & 8 & 4 \\ 4 & 3 & 2 & 1 & 2 \\ 8 & 7 & 6 & 5 & 3 \\ 1 & 5 & 3 & 7 & 6 \end{pmatrix}$$

For finding the mean, take the sum of the table contents and multiply by 1 / 9

1/9 *

1	1	1
1	1	1
1	1	1

2	1	2
6	5	3
3	7	6

$$1/9*[1*2+1*1+1*2+1*6+1*5+1*3+1*3+1*7+1*6] = 35/9=3.88= 4$$

Pixel value at position 3x3 can be changed to 4

Problem-2

A 4x4 image patch is shown below. Compute the value of the pixel at the position (2x2) if it is smoothened by a 2x2 average filter.

$$\begin{pmatrix} 0 & 1 & 2 & 3 \\ 5 & 6 & 7 & 8 \\ 4 & 3 & 2 & 1 \\ 8 & 7 & 6 & 5 \end{pmatrix}$$

Pixel value at the position (2*2) = 2

Here its 2x2, so multiply by 1/4

1/4

1	1
1	1

2	1
6	5

1/4[1*2+1*1+1*6+1*5]= ¼[14] = 7

So the pixel value 2 is changed to 7

---

4. Edge detection operators-Prewitt and Sobel operator

Sobel Operator

What is Sobel Operator?

- Its an operator used for edge detection
  - For Finding the edges in an image
- Its a discrere differentiation operator
  - Think of it as a tool that helps to find where there are changes in brightness in an image. These changes often correspond to edges of objects in the image.
- It computes the gradient approximation of image density function for image edge detection

    - In an image, the gradient represents the rate at which the brightness changes from one pixel to another. The Sobel Operator helps to estimate this gradient.

How does Sobel Operator work?

- The Sobel Operator uses two small 3x3 grids (called kernels or masks) to process the image. These grids are used to calculate the change in brightness in two perpendicular directions: horizontal and vertical.
- **Kernels (Masks)**:
    - These are small matrices that you slide over the image, pixel by pixel, and perform calculations to detect changes in brightness.
- **Two 3x3 Kernels**:
    - One kernel is used to detect changes in the horizontal direction
    - The other kernel is used to detect changes in the vertical direction

Advantages of Sobel

- Simple and time efficient
- Very easy at searching smooth edges

Limitations of Sobel

- HIghly sensitive to noise
- Not every accurate at edge detection
- Thick and rough edges does not give appropriate results

Prewitt Operator

What is Prewitt Operator?

- This operator is almost similar to sobel operator
- This also detects vertical and horizontal edges of an image
- Its one of the best ways to detect the orientation and magnitude of an image
- It uses kernels or masks

Advantages of Prewitt

- Good performance detecting vertical and horizontal edges
- Best operator to detect the orientation of an image

Limitations of Prewitt

- The magnitude of coefficient is fixed and cannot be changed
- Diagonal direction points are not preserved always

5. Image segmention

What is image segmentation

- Image Segmentation is the divison of an image into regions or categories, which correspond to different objects or parts of objects
- Every pixel in an image is allocated to one of a number of these categories

Conditions for good segmentation

- Pixels in the same category have similar grey scale of multivariate values and form a connected region
- Neighbouring pixels which are in different categories have dissimilar values

Ways to partition

1. Partition an image based on abrupt changes in intesnity such as edges
    1. Example: Edge Detection
2. Partition an image into region that are similar according to a set of predefined criteria
    1. Examples:
        1. Thresholding, Region growing, region spitting, merging

6. Thresholding(Intensity, Global)

What is thresholding

- Image thresholding segmentation is a simple form of image segmentation
- Thresholding is a technique in image processing used to create simpler images by turning a grayscale image into a binary image, or segmenting it into different regions based on pixel intensity values.

How does thresholding work?

1. **Image Segmentation**:
    1. Image thresholding is a basic form of image segmentation, which is the process of dividing an image into different parts or regions.
2. **Setting a Threshold**:
    1. You set a specific threshold value, which is a cutoff point for pixel intensities in the image.
3. **Binary Image Creation**:
    1. Pixels with intensity values below or equal to the threshold are set to one value (e.g., black or 0).
    2. Pixels with intensity values above the threshold are set to another value (e.g., white or 255).
4. **Example**
    1. Threshold = 60
    2. Any pixel with an intensity of 60 or less becomes black (0).
    3. Any pixel with an intensity above 60 becomes white (255).
5. **Practical use**
    1. For an image with a clear distinction between the background and an object, thresholding helps to separate the object from the background based on their pixel intensities.
    2. A well-chosen threshold value is crucial for effective segmentation.

Intensity Thresholding

- Intensity thresholding is a technique used in image processing to separate objects from the background based on their intensity (brightness) values.

How Does Intensity Thresholding Work?

- **Image Composition**:
    - - Consider an image f(x,y)f(x,y) where the intensity of the pixels can range from dark to light.
    - - The image consists of light objects on a dark background, meaning the objects are brighter than the background.
- **Intensity Modes**:

- In this image, the pixel intensity values are grouped into two dominant modes
- One mode represents the intensity values of the background pixels (darker).
- The other mode represents the intensity values of the object pixels (lighter).

- **Selecting a Threshold T**
  - - To extract (separate) the objects from the background, choose a threshold value T.
  - This threshold TT is chosen such that it effectively divides the intensity values into two groups: one for the objects and one for the background.
- **Applying the Threshold**
  - **Object Points**:
    - Any pixel at position (x,y) in the image where the intensity value f(x,y) is greater than or equal to the threshold T is considered an object point.
  - **Background Points**:
    - - any pixel where the intensity value f(x,y) is less than the threshold T is considered a background point.

Global Thresholding

What is global thresholding

- Global thresholding is a method used in image processing to separate objects from the background by applying a single threshold value to the entire image.
- Key concepts
  - **Bimodal Image**:
    - A bimodal image has two distinct peaks in its intensity distribution plot (histogram).
    - One peak represents the intensity values of the object pixels.
    - The other peak represents the intensity values of the background pixels.
  - **Intensity Distribution Plot**:
    - This plot shows how frequently each intensity value occurs in the image.
    - For a bimodal image, you'll see two prominent peaks: one for the object and one for the background.

How Global Thresholding works

Determine the Threshold T

- Analyze the histogram of the image to identify the two peaks.
- Choose a threshold value T that lies between these two peaks to separate the object from the background.

Apply the Threshold:

- For the entire image, apply the same threshold T to every pixel.
- Any pixel with an intensity value greater than or equal to TT is considered part of the object.
- Any pixel with an intensity value less than T is considered part of the background.

Example

- Imagine an image with pixel intensities ranging from 0 (black) to 255 (white):
- The histogram shows two peaks: one around intensity 50 (background) and another around intensity 200 (object).
- Choose a threshold T of 125.
  - - Pixels with intensity ≥125 are labeled as object points (e.g., set to white in a binary image).
  - Pixels with intensity <125 are labeled as background points (e.g., set to black in a binary image).

Advantages

- **Simplicity**: Easy to implement and computationally efficient.
- **Effective for Bimodal Images**: Works well when there is a clear distinction between the object and background intensities.

Disadvantages

- **Poor Performance with Poor Illumination**:
  - If the image has uneven lighting or shadows, the intensity values may not be distinctly bimodal.
  - The single threshold T may not effectively separate the object from the background in all regions of the image.

---

7. Region growing, splitting, merging

What is a region?

- A **region** refers to a group of connected pixels that share similar properties, such as intensity, color
- **Characteristics of a Region**
    - **Connected Pixels**:
        - Pixels in a region are connected to each other, meaning they form a contiguous area within the image.
    - **Similarity in Properties**:
        - The pixels in a region have similar attributes. These attributes can include:
        - - **Intensity**: Brightness levels of the pixels.
        - - **Color**: Specific color values of the pixels.
- **Region based segmentation**
    - Region-based segmentation is a method used to divide an image into meaningful regions.
    - This method is different from edge-based segmentation, which focuses on finding the boundaries between different regions. Instead, region-based segmentation groups pixels into regions based on their similarity.

Region Growing technique

What is Region Growing Technique?

- The region growing technique is a method used in image segmentation to group pixels into regions based on predefined criteria, starting from seed pixels and expanding outward.

How does Region Growing technique work?

- **Seed Pixel**:
    - - The process begins with selecting one or more initial pixels called seed pixels.
- **Checking Adjacent Pixels**:
    - - From the seed pixel, the algorithm examines adjacent pixels to see if they meet the predefined criteria.
- **Adding to the Region**:
    - - If an adjacent pixel meets the criteria, it is added to the region of the seed pixel.

    - This process continues iteratively, examining the neighbors of the newly added pixels and adding them to the region if they also meet the criteria.
- **Stopping Condition**:
    - The process stops when no more adjacent pixels meet the criteria, meaning the region cannot grow any further.

Criteria for Region Growing

- The predefined rules or criteria for adding pixels to a region are often based on a threshold. This threshold determines the similarity required for pixels to be part of the same region.

Example of Region Growing

- **Seed Pixel**: Let's start with a seed pixel having a value of 2.
- **Threshold Value**: Set a threshold value of 3.
- if a pixel has a value less than 3, then it willl be considered inside the seed pixel region. Otherwise it will be considered in another region.
- Hence 2 regions are formed in the following image based on a threshold value of 3

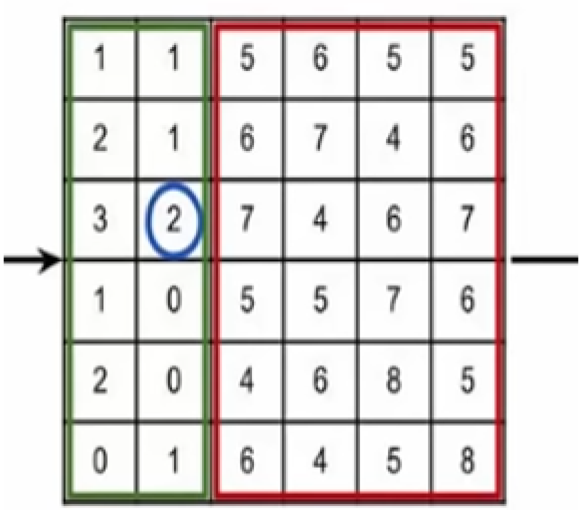
This is the original image, Taking 2 as seed

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

Original Image

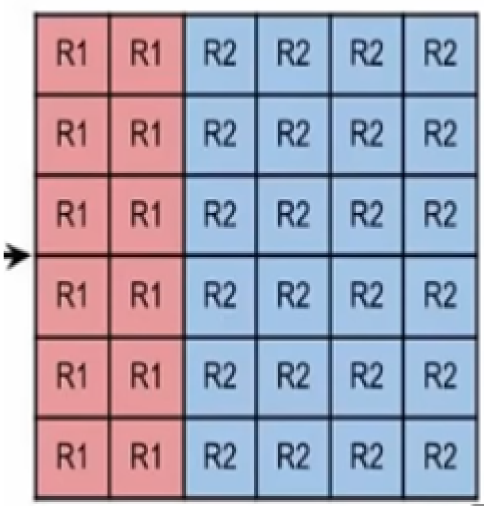
- Consider the neighbouring pixels of the seed pixel (2), Putting seed Pixel in Region 1
- Consider 1
    - 1 is less than threshold (3)

        - Put it in Region 1 (R1)
- Consider 2
    - 2 is less than threshold, adding to R1
- Repeat for other values
- Consider 7
    - 7 is greater than threshold (3)
        - Put it outside Region 1
        - So putting it in Region 2
- Consider 5
    - 5 > threshold
    - Putting it in Region 2
- Repeat for other values



Region growing process with 2 as the seed pixel.

Here Green is Region 1 and Red is Region 2



Splitting image into two regions based on a threshold.

Region Splitting and Merging technique

Region Splitting

- **Initial Step**:
    -
        - The entire image is initially considered as a single region.
- **Predefined Rules**:
    - These rules are criteria that define whether a region is homogeneous (similar in terms of intensity, color, texture, etc.). If a region does not meet these rules, it needs to be split.
- **Splitting Process**:
    - If the region does not follow the predefined rules, it is divided into smaller regions, usually into four quadrants.
    - The predefined rules are then applied to each of these smaller regions to check for homogeneity.
    - This process continues recursively, splitting regions until all resulting regions meet the predefined rules and are considered homogeneous.

Region Merging

- **Initial Step**:
    - Start by considering each pixel or a very small region (e.g., 2x2 pixels) as an individual region.
- **Seed Region**:

- Select a seed region to start with.

- **Merging Process**:
  - Check the adjacent regions to see if they meet the predefined similarity rules.
  - If the adjacent regions are similar, merge them into a single region.
  - Continue this process, checking and merging adjacent regions, until no further merging is possible.

Combining Splitting and Merging

- **Splitting Phase**:
  - - First, the entire image is split into the smallest possible regions where each region follows the predefined rules.
  - - This ensures that the image is divided into many small homogeneous regions.
- **Merging Phase**:
  - - After splitting, the regions are checked for similarity based on the predefined rules.
  - - Similar adjacent regions are then merged to form larger homogeneous regions.

Example

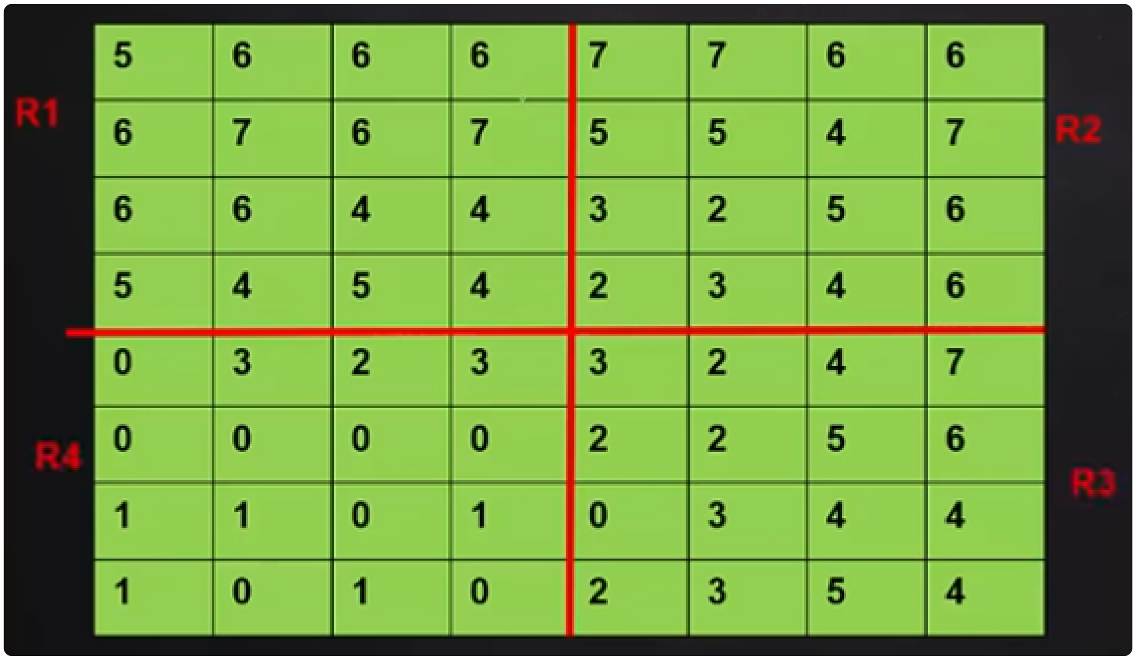
Splitting

Apply region splitting on the following image. Assume the threshold value be<=4.

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

- Take difference between Max and Minimum in the above table
  - Here max value is 7
  - Min value is 0

    - 7 -0 = 7

- In the question its given that the threshold value is less than 4
    - But here 7 - 0 = 7 >4
    - Its greater than 4
    - **So we will split this into 4 quadrants**
- 

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

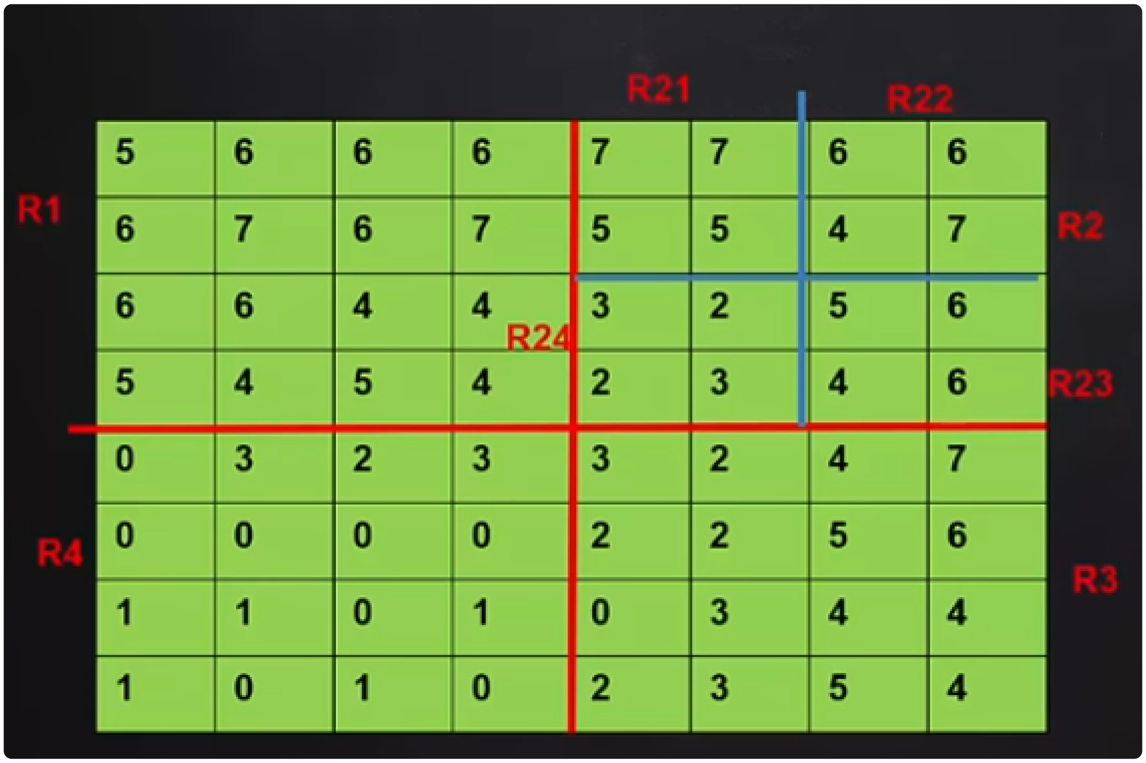
- After splitting, there are 4 regions, R1, R2, R3, R4
- Lets take each region one by one

Region 1

- In Region 1 we will go through the numbers in that area and get max and min values
- Max value = 7
- Min Value = 3
- 7 - 4 = 3 <= Threshold (4)
- No need of splitting

Region 2

- Max value = 7
- Min Value = 2
- 7 - 2 = 5 > Threshold (4)
- Splitting needed
- Divide R2 to 4 quadrants



Region R21

- 7-5 = 2 < 4, No Split

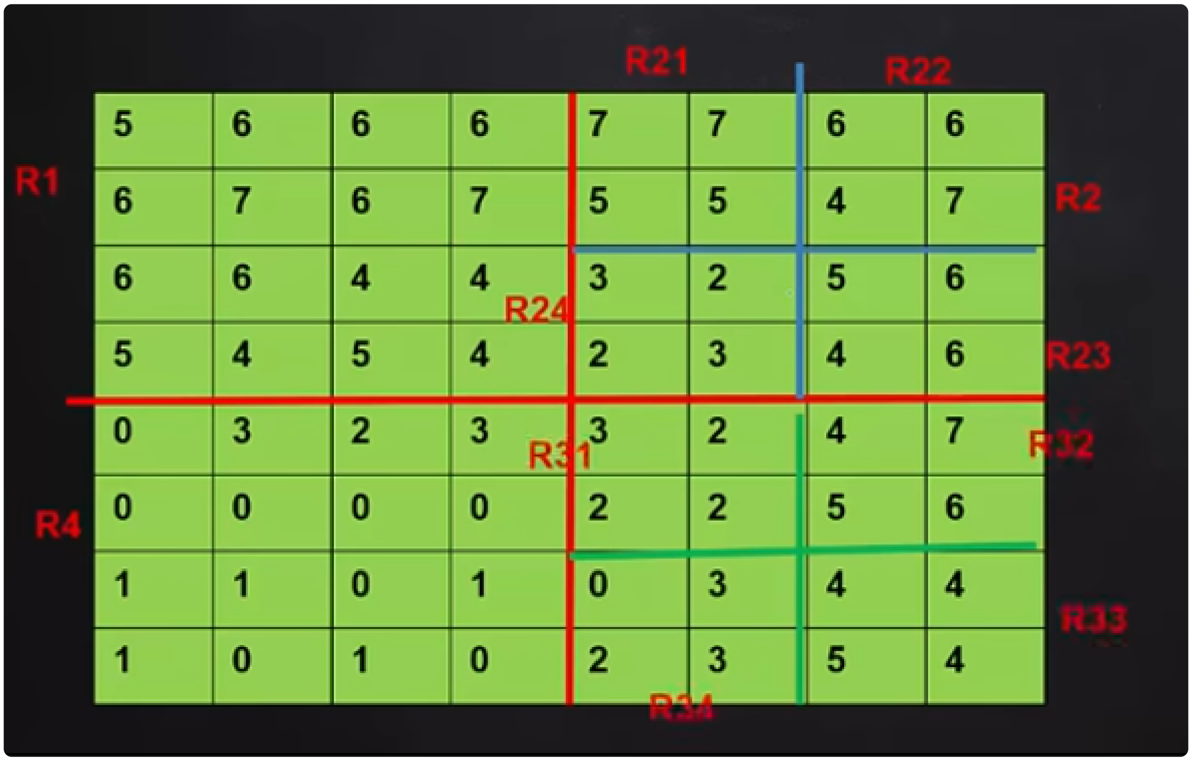
  Region R22
- 7-4 = 3 < 4, No Split

  Region R23
- 6-4 = 2 < 4, No Split

  Region R24
- 3-2 = 1 < 4, No Split

Region 3

- 7-0 = 7 >4
  - Splitting needed



Region R31

- 3-2 = 1 < 4, No Split

  Region R32
- 7-4 = 3 < 4, No Split

  Region R33
- 5-4 = 1 < 4, No Split

  Region R34
- 3-0 = 3 < 4, No Split

Region 4

- 3 -0 = 3 < 4 No split

Merging

- **R1,R21**
  - We need to check both directions, Max of R1 - Min of R21 and Max of R21 - Min of R
    - And verify whether the result of both are below the threshold
    - If both are below threshold, then it can be merged
  - **R1 and R21**

    - R1 Max-7

- R21 Min 5

- 7 - 5 = 2 <=4

- Less than threshold

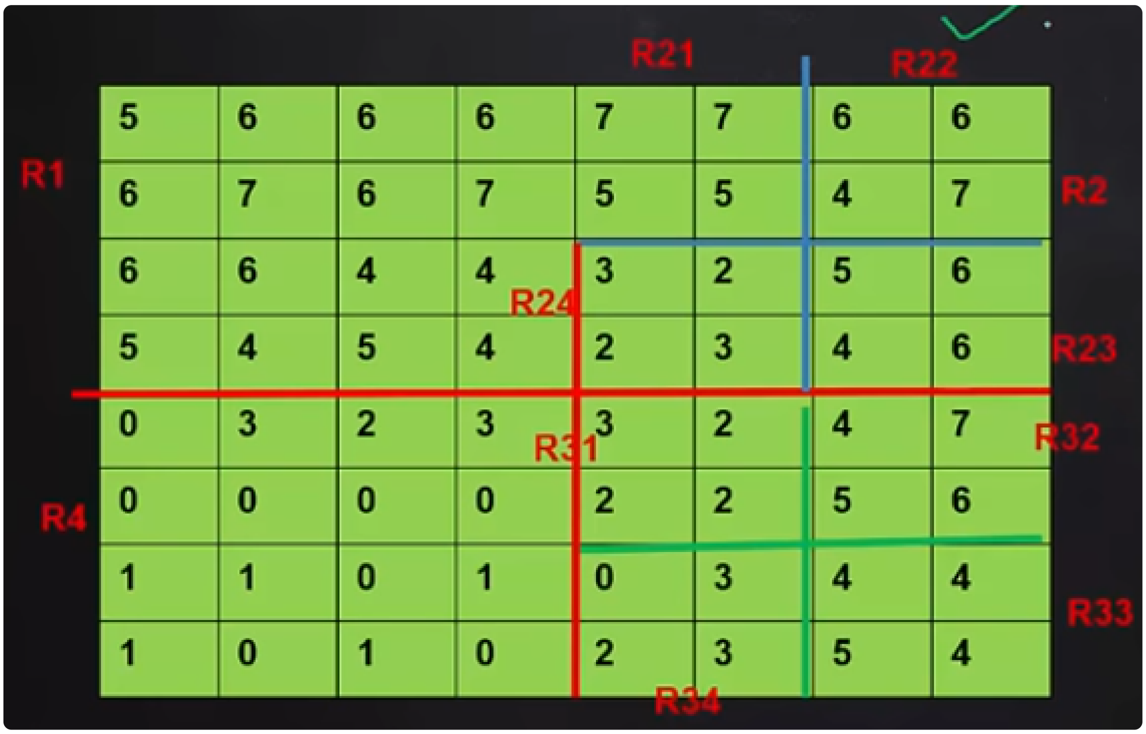
R21 and R1

- R21 Max = 7

- R Min 4

- 7-4 = 3 <=4

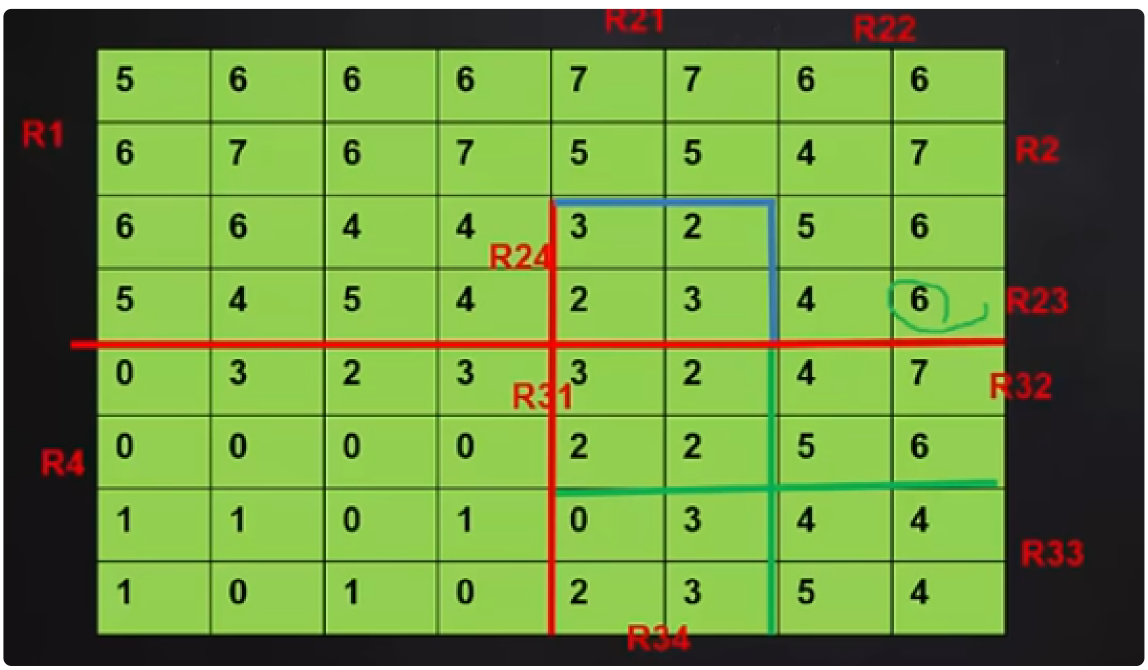
- Less than threshold

    - Both are less than threshold, can be merged
        - 

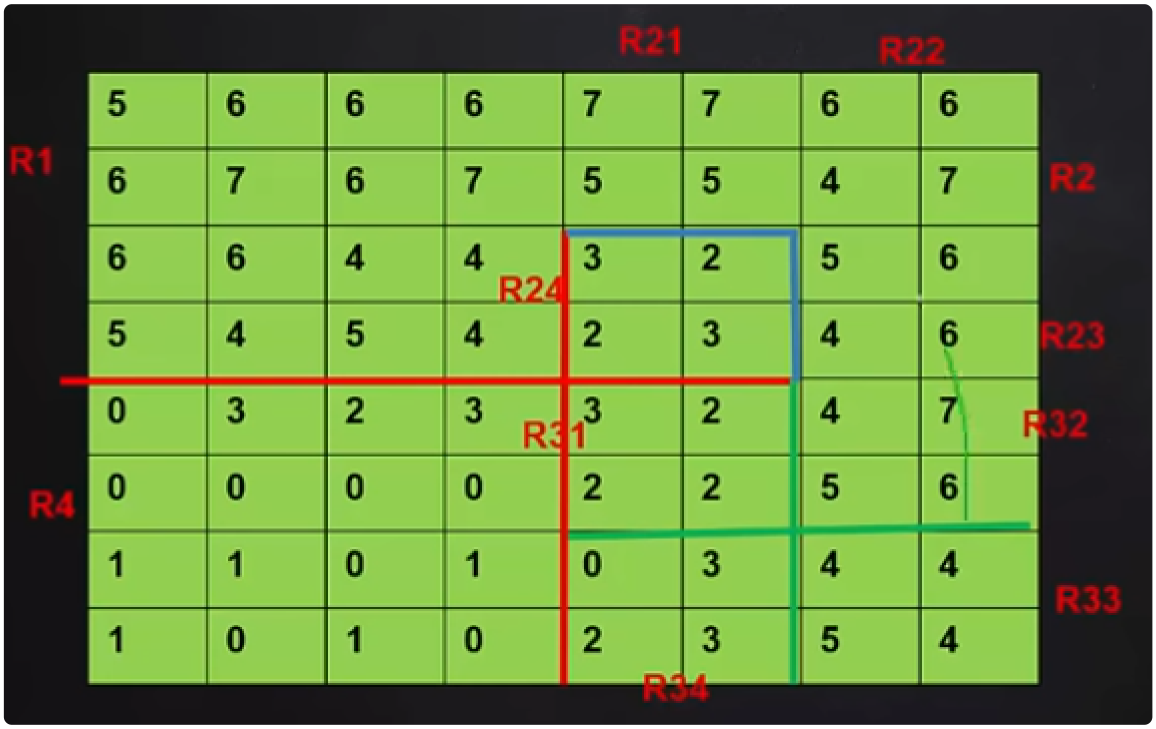
- **R1-R21, R22**
    - **R1-R21 and R22**
        - 7 - 4 = 3, Less than threshold
    - **R22 and R1-R21**
        - 7 - 4 = 3 less than threshold
    - Merging

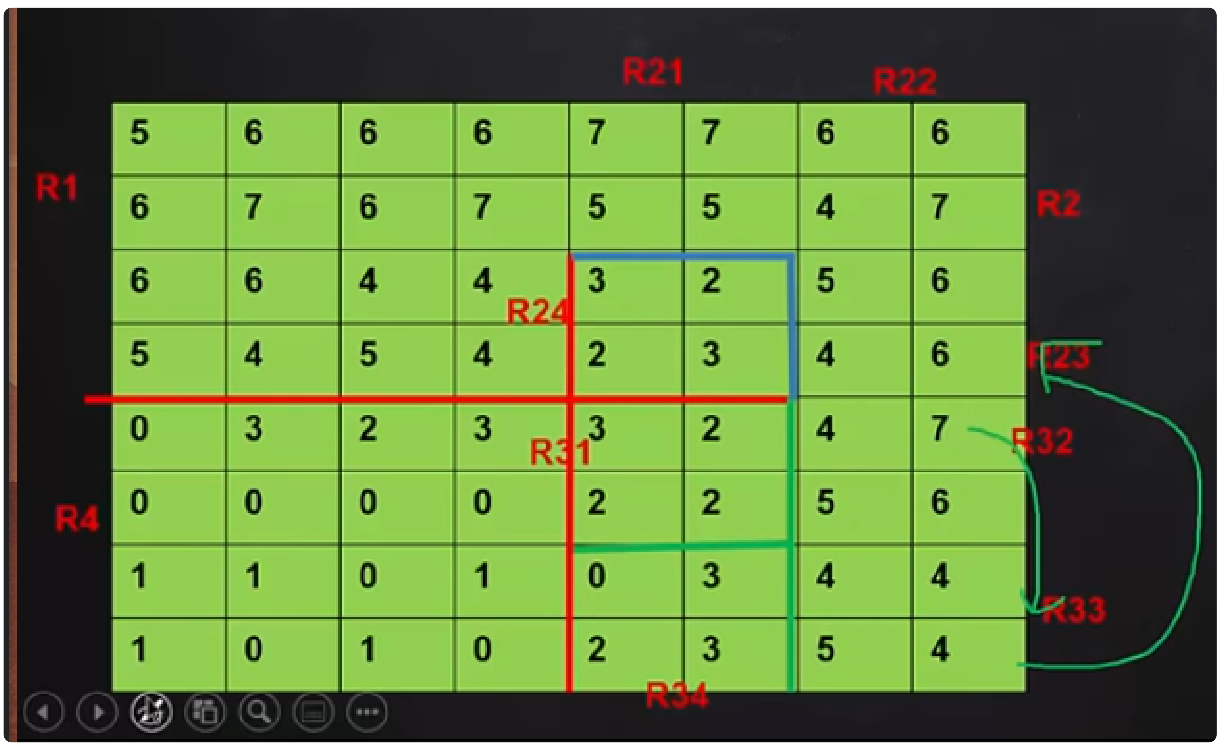
- 

- **R1-R21-R22, R23**
    - Merging

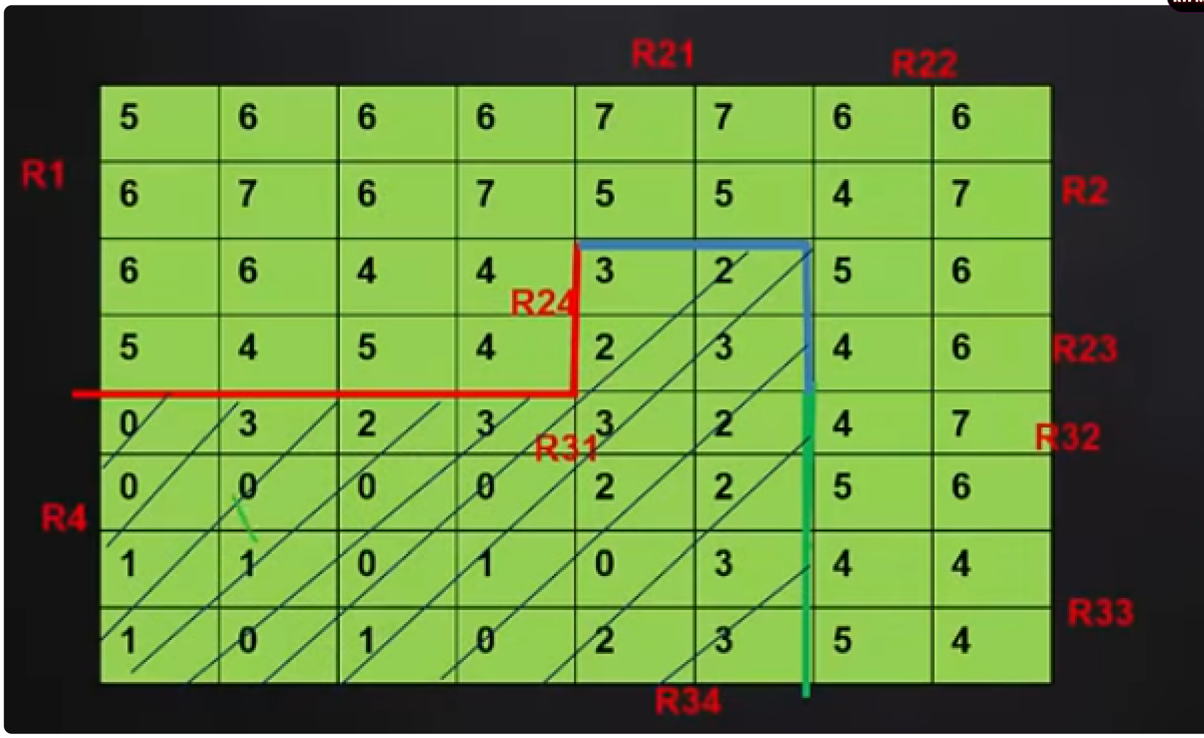
- 

- **R1-R21-R22-R23, R32**
    - Merging

- 

- **R1-R21-R22-R23-R32,R33**
  - Merging
  - 

- **R1-R21-R22-R23-R32-R33,R34**
  - Not merging
    - 7 - 2 = 5, > threshold, condition fails
- **After repeating these steps, we get 2 regions**

- 
R21
R22
R1
R2
R24
R23
R31
R32
R4
R33
R34
5 6 6 6 7 7 6 6
6 7 6 7 5 5 4 7
6 6 4 4 3 2 5 6
5 4 5 4 2 3 4 6
0 3 2 3 3 2 4 7
0 0 0 0 2 2 5 6
1 1 0 1 0 3 4 4
1 0 1 0 2 3 5 4