Counting Sort Algorithm:

This is one of the techniques to sort the array elements; it performs sorting by calculating the number of occurrences of a particular element in an array. It maintains a separate array to count the number of occurrences of the element; after that, it performs sorting by mapping the count.

As we have already known that it is a sorting algorithm based on the counting of the unique element.

1) Start

2) Count array to maintain the count of the unique elements inside the input.

3) Modify the count array to contain the sum of the previous counts.

4) map data the array and decrees the count by 1.

Time Complexity: **O(n+k)**

Quick Sort Algorithm:

Quick Sort is a Divide and Conquer algorithm. It picks an element as pivot and partitions the given array around the picked pivot.

Technically, quick sort follows the below steps:
Step 1 − Make any element as pivot
Step 2 − Partition the array on the basis of pivot
Step 3 − Apply quick sort on left partition recursively
Step 4 − Apply quick sort on right partition recursively

Time Complexity: **Best case** $= \Omega(\boldsymbol{nlog}(\boldsymbol{n}))$**, Average** $= \boldsymbol{\theta}(\boldsymbol{nlog}(\boldsymbol{n}))$**,**
Worst case $= \boldsymbol{O}(\boldsymbol{n}^{\boldsymbol{2}})$

WHICH ALGORITHM PREFORM BETTER IN EACH FILTER?

Adaptive :

In adaptive-median filter: CountingsSort preform better because it consumes a little time.and IT'S Time Complexity: n+k

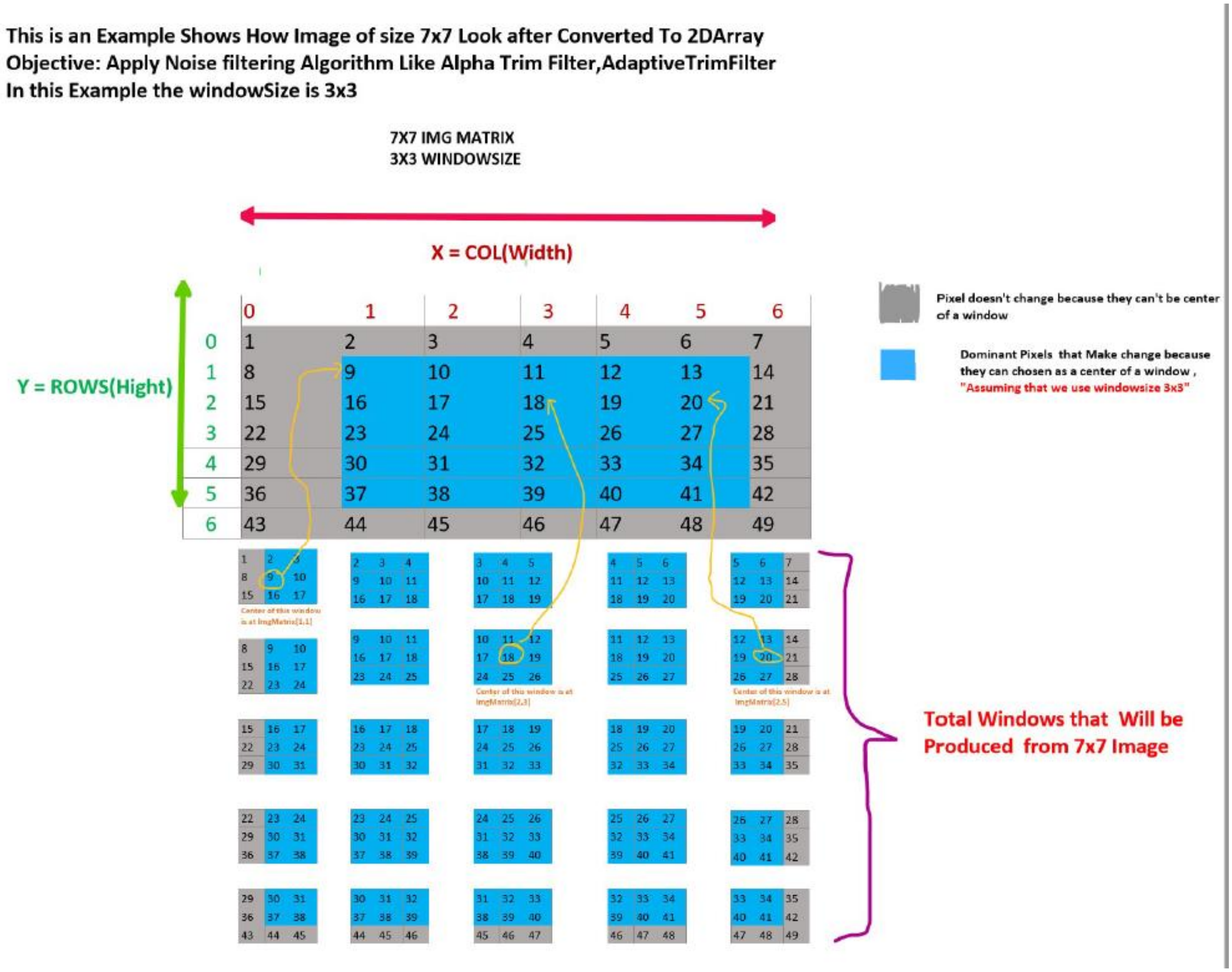
While quicksort Time Complexity: n^2

Alpha :

In Alpha-Trim filter: KthElement preform better because it doesn't sort at all but it find the max and min then calculate the avg value

While CoutningSort More time due to sorting Process

Our Approach To To Filter an image:



This is an Example Shows How Image of size 7x7 Look after Converted To 2DArray
Objective: Apply Noise filtering Algorithm Like Alpha Trim Filter,AdaptiveTrimFilter
In this Example the windowSize is 3x3

7X7 IMG MATRIX
3X3 WINDOWSIZE

X = COL(Width)

Y = ROWS(Hight)

	0	1	2	3	4	5	6
0	1	2	3	4	5	6	7
1	8	9	10	11	12	13	14
2	15	16	17	18	19	20	21
3	22	23	24	25	26	27	28
4	29	30	31	32	33	34	35
5	36	37	38	39	40	41	42
6	43	44	45	46	47	48	49

▨ Pixel doesn't change because they can't be center of a window

🟦 Dominant Pixels that Make change because they can chosen as a center of a window , "Assuming that we use windowsize 3x3"

1	2	3	2	3	4	3	4	5	4	5	6	5	6	7
8	9	10	9	10	11	10	11	12	11	12	13	12	13	14
15	16	17	16	17	18	17	18	19	18	19	20	19	20	21
8	9	10	9	10	11	10	11	12	11	12	13	12	13	14
15	16	17	16	17	18	17	18	19	18	19	20	19	20	21
22	23	24	23	24	25	24	25	26	25	26	27	26	27	28
15	16	17	16	17	18	17	18	19	18	19	20	19	20	21
22	23	24	23	24	25	24	25	26	25	26	27	26	27	28
29	30	31	30	31	32	31	32	33	32	33	34	33	34	35
22	23	24	23	24	25	24	25	26	25	26	27	26	27	28
29	30	31	30	31	32	31	32	33	32	33	34	33	34	35
36	37	38	37	38	39	38	39	40	39	40	41	40	41	42
29	30	31	30	31	32	31	32	33	32	33	34	33	34	35
36	37	38	37	38	39	38	39	40	39	40	41	40	41	42
43	44	45	44	45	46	45	46	47	46	47	48	47	48	49

Center of this window is at ImgMatrix[1,1]

Center of this window is at ImgMatrix[2,3]

Center of this window is at ImgMatrix[2,5]

Total Windows that Will be Produced from 7x7 Image

Common Steps between both Filters: Alpha,Adaptive :

1- Get The ImgMaterix2d Array

2-make nested loop to get Each Pixel in the 2darray

3-Check if This Pixel Can be Starting Point of An Window of Size ,if no get next pixel, How to check?

If pixel at (y,x) can be initial pixel for creating a window
Then these Condition Must Exsit
initial X +(windowSize-1) < img.Width
initial Y +(windowSize-1) < img.Hight

4 -get y,x of the center pixel of selctedWindow , How ?

To Pick Center:
Int CenterpixelY = CurrentpixelY + winSize/2
Int CenterpixelX= CurrentpixelX + winSize/2

5- get the items inside the windowSize , How ?

To Pick All Item in window:
1)create 1d array to append values inside it
2)Since we have the initial y, initial x for example let our current pixel be the first pixel , then yi =0,xi=0

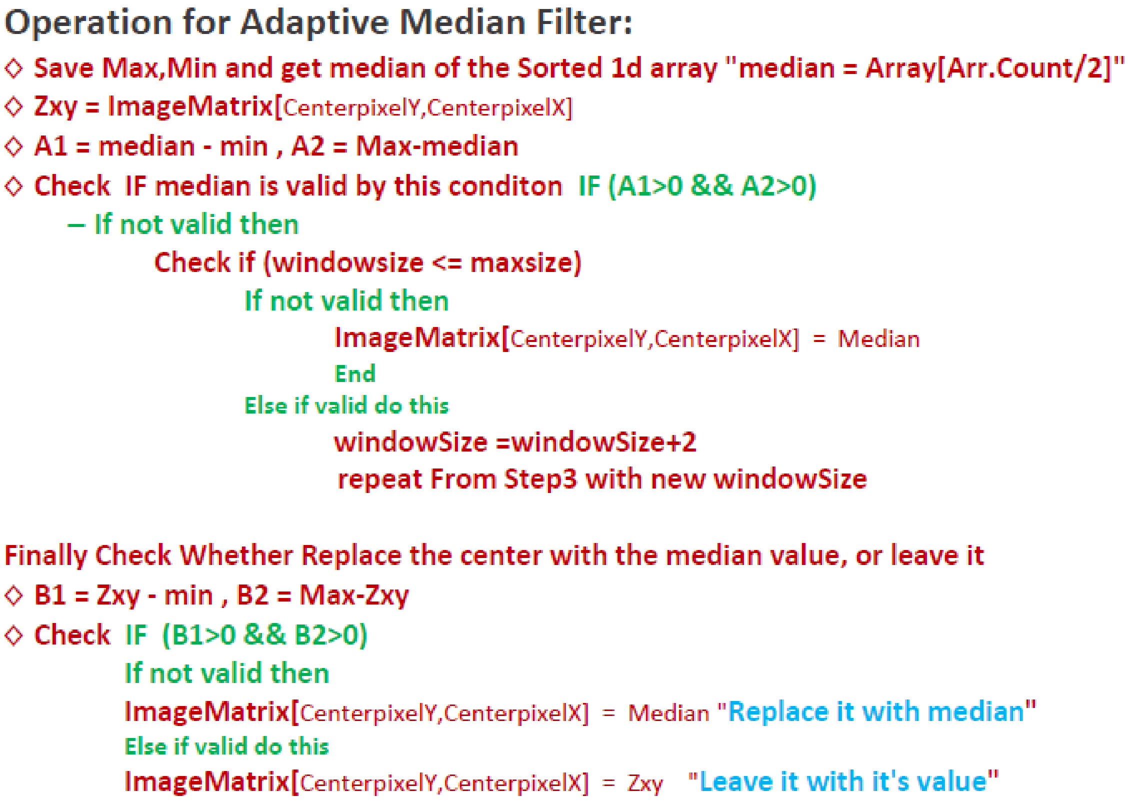
We need to make nested loop to get these values in the 2darray and append them in our 1darray
ImgMatrix[0,0] ImgMatrix [0,1] ImgMatrix [0,2]
ImgMatrix[1,0] ImgMatrix [1,1] ImgMatrix [1,2]
ImgMatrix[1,0] ImgMatrix [1,1] ImgMatrix [1,2]

6- Sort the array using any sorting algorithm that Is required

7-Make Some Operation to get a newpixel value

Operation for Alpha Trim Filter:

1) remove Max,Min Values in the Sorted 1d array
2) Calculate ther Average
3) ImageMatrix[CenterpixelY,CenterpixelX] **= Average**



Operation for Adaptive Median Filter:

```
◇ Save Max,Min and get median of the Sorted 1d array "median = Array[Arr.Count/2]"
◇ Zxy = ImageMatrix[CenterpixelY,CenterpixelX]
◇ A1 = median - min , A2 = Max-median
◇ Check  IF median is valid by this conditon  IF (A1>0 && A2>0)
    – If not valid then
          Check if (windowsize <= maxsize)
                If not valid then
                      ImageMatrix[CenterpixelY,CenterpixelX]  =  Median
                      End
                Else if valid do this
                      windowSize =windowSize+2
                       repeat From Step3 with new windowSize

Finally Check Whether Replace the center with the median value, or leave it
◇ B1 = Zxy - min , B2 = Max-Zxy
◇ Check  IF  (B1>0 && B2>0)
        If not valid then
        ImageMatrix[CenterpixelY,CenterpixelX]  =  Median "Replace it with median"
        Else if valid do this
        ImageMatrix[CenterpixelY,CenterpixelX]  =  Zxy    "Leave it with it's value"
```