In [1]:

```
import cv2
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import random
from scipy import signal
from scipy import misc
from skimage import color
from skimage import io
```

Part 1

In [2]:

```
X = cv2.imread('5_1.bmp')
print(X.shape)
original = cv2.imread('5_1.bmp')

Xhsv = cv2.cvtColor(X, cv2.COLOR_BGR2HSV)

img = color.rgb2gray(io.imread('5_1.bmp'))
print(img.shape)

black = 0
white = 0

counter = 0
for row in range(0, 512, 1):
    for col in range(0, 512, 1):
        probability = 0.15
        x = random.uniform(0, 1)
        if x <= probability:
            if black >= white:
                Xhsv[row, col, 2] = 255
                white = white + 1
            else:
                Xhsv[row, col, 2] = 0
                black = black + 1


saltandpepper = cv2.cvtColor(Xhsv, cv2.COLOR_HSV2BGR)

lowpassfilter = 1/9 * np.array([[1, 1, 1],
                               [1, 1, 1],
                               [1, 1, 1]])



#lowpassimage = signal.convolve2d(Xhsv[:, :, 2], lowpassfilter)
lowpassimage = cv2.filter2D(Xhsv[:, :, 2], -1, lowpassfilter)

cv2.imshow('original', original)
cv2.imshow('salt and pepper noisy image', saltandpepper)
cv2.imshow('low pass image', lowpassimage)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

```
(512, 512, 3)
(512, 512)
```

In [3]:

```
def median(twodimage, windowsize):
    row, col  = twodimage.shape
    paddingvalue = int((windowsize - 1) / 2)
    paddedarray = cv2.copyMakeBorder(twodimage, paddingvalue, paddingvalue, paddingvalue, paddingvalue, cv2.BORDER_REPLICATE)

    result = np.zeros((row, col))

    for i in range(row):
        for j in range(col):
            array = np.zeros(windowsize**2)
            for x in range(0, windowsize):
                for y in range(0, windowsize):
                    value = paddedarray[i + x, j + y]
                    array[int(x * windowsize + y)] = value
            median = np.median(array)
            result[i, j] = median
    return result
```

In [5]:

```
X = cv2.imread('5_1.bmp')
print(X.shape)
original = cv2.imread('5_1.bmp')

Xhsv[:, :, 2] = median(Xhsv[:, :, 2], 5)

medianpassimagefinal = cv2.cvtColor(Xhsv, cv2.COLOR_HSV2BGR)
cv2.imshow('original', X)
cv2.imshow('median pass image', medianpassimagefinal)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

```
(512, 512, 3)
```

Part Two

In [9]:

```
Y = cv2.imread('5_2.bmp')
Yhsv = cv2.cvtColor(Y, cv2.COLOR_BGR2HSV)
Yhsv[:, :, 2] = highboost(1, Yhsv[:, :, 2])
final = cv2.cvtColor(Yhsv, cv2.COLOR_HSV2BGR)
cv2.imshow('original', Y)
cv2.imshow('high boosted image', final)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

In [8]:

```
def highboost(A, vchannel):
    row, col = vchannel.shape
    highboostfilter = np.array([[-1, -1, -1],
                                [-1, A + 8, -1],
                                [-1, -1, -1]])
    highpassimagearray = cv2.filter2D(vchannel, -1, highboostfilter)
    return highpassimagearray
```

In [ ]: