Problem Statement:

Read any image. Display the histogram, Equalized histogram, and image with equalized histogram.

In [1]:
```
%cd ..
```

```
/
```

In [17]:
```
import matplotlib.pyplot as plt
import cv2
from google.colab.patches import cv2_imshow
```

In [18]:
```
image_path = 'bird_image.jpeg'

image = cv2.imread(image_path)
cv2_imshow(image)
```

Displaying Histogram for the above image:

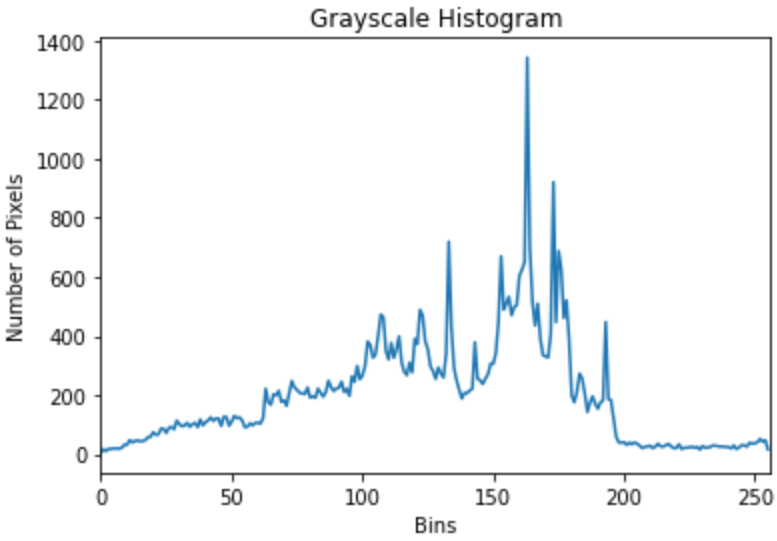
This code will load the image, convert it to grayscale, and calculate the histogram of the grayscale image. It will then plot the histogram using matplotlib. The histogram will have 256 bins, corresponding to the possible gray levels in the image.

In [19]:
```
# Load the image and convert it to grayscale
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

cv2_imshow(gray)
```

In [20]:
```
# Calculate the histogram
hist = cv2.calcHist([gray], [0], None, [256], [0, 256])

# Plot the histogram
plt.figure()
plt.title("Grayscale Histogram")
plt.xlabel("Bins")
plt.ylabel("Number of Pixels")
plt.plot(hist)
plt.xlim([0, 256])
plt.show()
```



Displaying Equalized Histogram for the image:

This code will equalize the histogram of the grayscale image using the cv2.equalizeHist() function, and calculate the histogram of the equalized image. It will then plot the histogram using matplotlib. The histogram will have 256 bins, corresponding to the possible gray levels in the image.

In [21]:
```
# Equalize the histogram
eq_hist = cv2.equalizeHist(gray)

# Calculate the histogram
hist2 = cv2.calcHist([eq_hist], [0], None, [256], [0, 256])

# Plot the histogram
plt.figure()
plt.title("Equalized Grayscale Histogram")
plt.xlabel("Bins")
```

```
plt.ylabel("Number of Pixels")
plt.plot(hist2)
plt.xlim([0, 256])
plt.show()
```

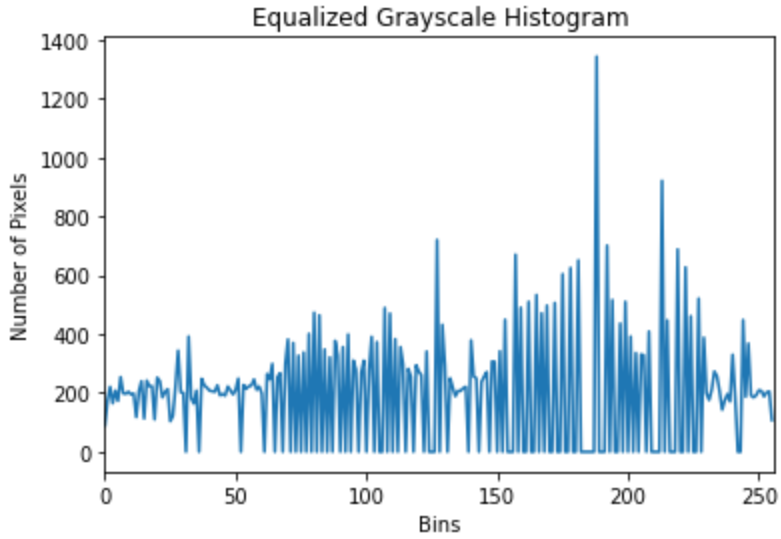


Image corresponding to the equalized histogram:

In [22]:
```
cv2_imshow(eq_hist)
```

Difference between the original grayscale image and the image corresponding to the equalized histogram:

In [25]:
```
cv2_imshow(gray)
cv2_imshow(eq_hist)
```

In [ ]: