Student Names and IDs:

- Matias Benitez, 2416556
- Usa Kerdnunvong, 1950448
- Neil Pruthi, 2419654

Homework 1

Part 1: Calculus

Problem 1.1 ¶

Write an expression for the gradient of f,

$$\nabla f(\mathbf{x}) = \frac{\partial f}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} .$$

Answer

$$\nabla f(\mathbf{x}) = \begin{bmatrix} 3x^2 - 6x \\ 3y^2 - 6y \end{bmatrix}$$

Problem 1.2

Write an expression for the Hessian of f,

$$H_f(\mathbf{x}) = \frac{\partial^2 f}{\partial \mathbf{x} \partial \mathbf{x}^T} = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix} .$$

Answer

$$H_f(\mathbf{x}) = \begin{bmatrix} 6x - 6 & 0 \\ 0 & 6y - 6 \end{bmatrix} .$$

Problem 1.3

Give expressions for the eigenvalues $\lambda_1(x, y)$ and $\lambda_2(x, y)$ (in any order) of H_f.

Answer

The eigenvalues $\lambda_1(x, y)$ and $\lambda_2(x, y)$ are the two values of λ that satisfy $\det(H_f - \lambda I) = 0$ where I is the 2×2 identity matrix. Solving, I get

$$\lambda_1(x, y) = 6x - 6$$
$$\lambda_2(x, y) = 6y - 6$$

Problem 1.4

Give expressions for all the unit-norm eigenvectors $\mathbf{u}_1$ of H_f corresponding to the eigenvalue $\lambda_1(x, y)$ you found in the previous problem and for all the unit-norm eigenvectors $\mathbf{u}_2$ corresponding to the eigenvalue $\lambda_2(x, y)$.

Answer

For $\lambda_i \in \{\lambda_1(x,y), \lambda_2(x,y)\}$, the corresponding eigenvector $\mathbf{v}_i$ satisfies $(H_f - \lambda_i I)\mathbf{v}_i = \mathbf{0}$. Solving, we have that

$$\mathbf{v}_1 = \begin{bmatrix} c_1 \\ 0 \end{bmatrix}$$
$$\mathbf{v}_2 = \begin{bmatrix} 0 \\ c_2 \end{bmatrix}$$

where $c_1, c_2 \in \mathbb{R}$. To get the expression of the unit-norm eigenvectors $\mathbf{u}_1$ corresponding to $\lambda_1(x,y)$ and $\mathbf{u}_2$ corresponding to $\lambda_2(x,y)$, I have

$$\mathbf{u}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$
$$\mathbf{u}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Problem 1.5

Give *all* the stationary points of f. For each of them, state whether it is a minimum, a maximum, or a saddle point, and give unit vectors along the directions of maximum and minimum curvature (or state that curvatures in all directions are equal, if that is the case).

Answer

Setting $\frac{\partial f}{\partial x} = 3x^2 - 6x = 0$ and $\frac{\partial f}{\partial y} = 3y^2 - 6y = 0$, and solving, I obtain the four stationary points $(0,0), (0,2), (2,0), (2,2)$.

For the curvature of each stationary point, I will use $\lambda_1(x,y) = 6x - 6$ and $\lambda_2(x,y) = 6y - 6$ from problem 1.3, and their corresponding eigenvectors

$$\mathbf{u}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$
$$\mathbf{u}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

The largest absolute value eigenvalue at each critical point is the amount of the curvature, and its respective eigenvector is the direction of the greatest curvature.

$(0, 0)$**:**

- maximum point, since $\frac{\partial^2 f(0,0)}{\partial x^2} \frac{\partial^2 f(0,0)}{\partial y^2} - \frac{\partial^2 f(0,0)}{\partial x \partial y} = (-6)(-6) - 0 = 36 > 0$ and $\frac{\partial^2 f(0,0)}{\partial x^2} = -6 < 0, \frac{\partial^2 f(0,0)}{\partial y^2} = -6 < 0$
- calulating the principal curvatures, $\lambda_1(0, 0) = -6$ and $\lambda_2(0, 0) = -6$. Since the principal curvatures are equal, the curvatures in all directions ($\mathbf{u}_1$ and $\mathbf{u}_2$) are equal

$(0, 2)$**:**

- saddle point, since $\frac{\partial^2 f(0,2)}{\partial x^2} \frac{\partial^2 f(0,2)}{\partial y^2} - \frac{\partial^2 f(0,2)}{\partial x \partial y} = (-6)(12 - 6) - 0 = -36 < 0$
- calulating the principal curvatures, $\lambda_1(0, 2) = -6$ and $\lambda_2(0, 2) = 12 - 6 = 6$. Since the absolute values of the principal curvatures are equal, the curvatures in all directions ($\mathbf{u}_1$ and $\mathbf{u}_2$) are equal.

$(2, 0)$**:**

- saddle point, since $\frac{\partial^2 f(2,0)}{\partial x^2} \frac{\partial^2 f(2,0)}{\partial y^2} - \frac{\partial^2 f(2,0)}{\partial x \partial y} = (12 - 6)(-6) - 0 = -36 < 0$
- calulating the principal curvatures, $\lambda_1(2, 0) = 12 - 6 = 6$ and $\lambda_2(2, 0) = -6$. Since the absolute values of the principal curvatures are equal, the curvatures in all directions ($\mathbf{u}_1$ and $\mathbf{u}_2$) are equal.

$(2, 2)$**:**

- minimum point, since $\frac{\partial^2 f(2,2)}{\partial x^2} \frac{\partial^2 f(2,2)}{\partial y^2} - \frac{\partial^2 f(2,2)}{\partial x \partial y} = (12 - 6)(12 - 6) - 0 = 36 > 0$ and $\frac{\partial^2 f(2,2)}{\partial x^2} = 12 - 6 = 6 > 0, \frac{\partial^2 f(2,2)}{\partial y^2} = 12 - 6 = 6 > 0$
- calulating the principal curvatures, $\lambda_1(2, 2) = 12 - 6 = 6$ and $\lambda_2(2, 2) = 12 - 6 = 6$. Since the absolute values of the principal curvatures are equal, the curvatures in all directions ($\mathbf{u}_1$ and $\mathbf{u}_2$) are equal.

Problem 1.6

To check that your answers are plausible, write Python 3 code that displays a plot of 30 iso-value contours of the function f for $-0.5 \leq x \leq 2.5$ and $-0.5 \leq y \leq 2.5$. Draw a red dot at every stationary point.

Answer

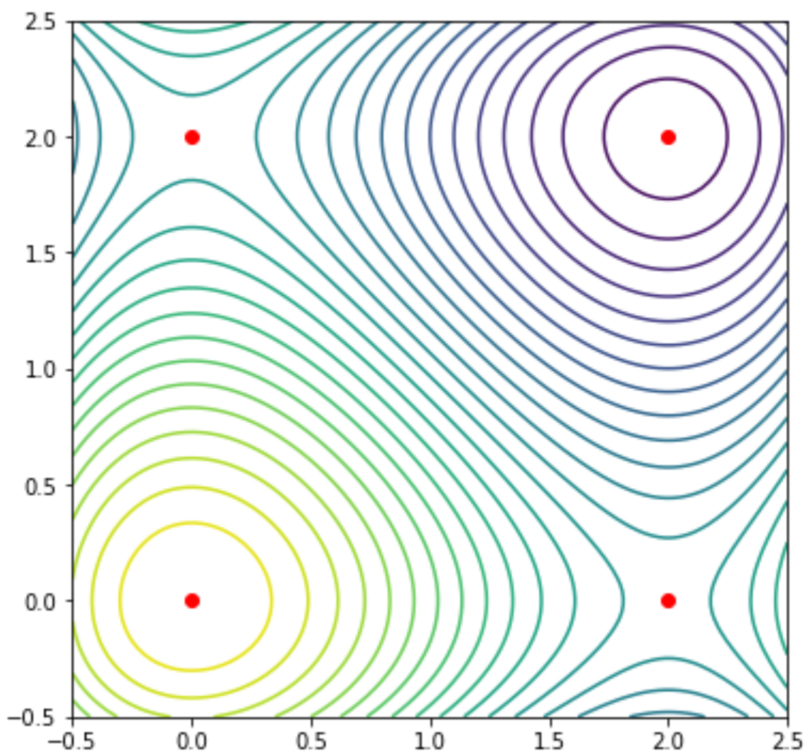
In [1]:
```
## code for creating graph

# importing packages
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

# code for points in the graph
ax_x = np.linspace(-0.5, 2.5, 101)
ax_y = np.linspace(-0.5, 2.5, 101)
x, y = np.meshgrid(ax_x, ax_y)
z = x**3 - 3 * x**2 + y**3 - 3 * y**2

# stationary points
crit_x = [0, 2, 0, 2]
crit_y = [0, 0, 2, 2]

# plot
plt.figure(figsize = (6,6))
plt.contour(x, y, z, 30)
plt.plot(crit_x, crit_y, 'ro')
plt.show()
```



Part 2: Linear Algebra

Problem 2.1

Give four matrices R, N, C, L whose *rows* are bases for the row space, null space, column space (a.k.a. range), and left null space of A. Use rows or columns of A for your basis vectors when possible, and vectors with integer entries in any case.

As a partial check, give the products AN^T and LA (fully spelled out with all their entries).

Answer

By doing the following elementary operations: $r_3 - r_1 - r_2$ and $r_3 - 2r_1$ we can transfor A to its echelom form:

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Therefore,

$$R = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \end{bmatrix}$$

In order to get the basis of the null space notice that $Ax = 0$ can be written by using the reduced form of A:

$X_1 + X_3 = 0$

$X_2 - X_3 = 0$

Therefore, x_3 is a free variable and the basis of the null space is

$$\begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix}$$

Thus,

$$N = \begin{bmatrix} -1 & 1 & 1 \end{bmatrix}$$

Since columns 1 and 2 have pivots in the reduced form of A, then:

$$\begin{bmatrix} 1 \\ 0 \\ 2 \\ 1 \end{bmatrix} . \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

are basis of the column space and,

$$C = \begin{bmatrix} 1 & 0 & 2 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

In order to get the left null space of A it is necessary to obtain the null space of A^T.

$A^T x = 0$

$\implies$

$$A^T = \begin{bmatrix} 1 & 0 & 2 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & -1 & 2 & 0 \end{bmatrix}$$

and its reduced form is:

$$\begin{bmatrix} 1 & 0 & 2 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

Therefore,

$$L = \begin{bmatrix} -2 & 0 & 1 & 0 \\ -1 & -1 & 0 & 1 \end{bmatrix}$$

Therefore, we finally get:

$$AN^T = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$LA = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Problem 2.2

Give the set S of *all* solutions to the system

$$A\mathbf{x} = \mathbf{0}$$

where A is the matrix above and $\mathbf{0}$ is a column vector of four zeros.

Answer

Since N contains the basis of the null space, then:

$$S = \{k\begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix} : k \in R\}$$

Problem 2.3

Give the set B of *all* vectors $\mathbf{b}$ for which the system

$$A\mathbf{x} = \mathbf{b}$$

admits a solution, if A is the matrix above.

Answer

Since C contains the basis of the column space, then:

$$B = \{k\begin{bmatrix} 1 \\ 0 \\ 2 \\ 1 \end{bmatrix} + s\begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} : k, s \in R\}$$

Part 3: Probability

Problem 3.1

What are the mean m_X and variance σ_X^2 of X?

Answer

$$
\begin{aligned}
m_X &= \sum_{i=0}^{2} \mathbb{P}(X = i) \times i \\
&= 0.3 \times 0 + 0.5 \times 1 + 0.2 \times 2 \\
&= 0.9 \\
\sigma_X^2 &= \sum_{i=0}^{2} \mathbb{P}(X = i) \times (i - m_X)^2 \\
&= 0.3 \times (0 - 0.9)^2 + 0.5 \times (1 - 0.9)^2 + 0.2 \times (2 - 0.9)^2 \\
&= 0.490
\end{aligned}
$$

In [2]:
```
## code for calculations
mean_x = 0.3 * 0 + 0.5 * 1 + 0.2 * 2
var_x = 0.3 * (0-0.9)**2 + 0.5 * (1 - 0.9)**2 + 0.2 * (2 - 0.9)**2
```

Problem 3.2

What is the mean m_Y of Y?

Answer

$$
\begin{aligned}
m_Y &= \sum_{i=0}^{2} \mathbb{P}(X = i) \times \mathbb{E}(Y \mid X = i) \\
&= 0.3 \times 6 + 0.5 \times 2 + 0.2 \times 5 \\
&= 3.8
\end{aligned}
$$

In [3]:
```
## code for calculations
mean_y = 0.3 * 6 + 0.5 * 2 + 0.2 * 5
```

Problem 3.3

What is the mean $m_{Y|X<2} = \mathbb{E}(Y \mid X < 2)$ of Y given that $X < 2$?

Answer

Given that $X < 2$, we know that there are now only two possible values of X, which are in the set $\{0, 1\}$. Therefore,

$$\begin{aligned} m_{Y|X<2} &= \frac{\mathbb{P}(X=0)}{\mathbb{P}(X=0)+\mathbb{P}(X=1)}\mathbb{E}(Y \mid X = 0) + \frac{\mathbb{P}(X=1)}{\mathbb{P}(X=0)+\mathbb{P}(X=1)}\mathbb{E}(Y \mid X = 1) \\ &= \frac{0.3}{0.3+0.5} \times 6 + \frac{0.5}{0.3+0.5} \times 2 \\ &= 3.50 \end{aligned}$$

In [4]:
```
## code for calculations
mean_ycond = 0.3 / 0.8 * 6 + 0.5 / 0.8 * 2
```

Part 4: Python 3

In [5]:
```
import numpy as np

across = ['marco', 'oneup', 'skate', 'elder', 'seeya']
puzzle = np.array([list(word) for word in across])
print(puzzle)
```

```
[['m' 'a' 'r' 'c' 'o']
 ['o' 'n' 'e' 'u' 'p']
 ['s' 'k' 'a' 't' 'e']
 ['e' 'l' 'd' 'e' 'r']
 ['s' 'e' 'e' 'y' 'a']]
```

Problem 4.1

Write a function with header

```
def prettyPrint(p):
```

that pretty-prints any *nonempty* array like `puzzle` according to the given specifications. Show your code and the result of pretty-printing the given array `puzzle`.

Answer

In [6]:
```
## function
def prettyPrint(p):
    assert p.ndim == 2, 'Puzzle must have two dimensions'
    for x in list(range(len(p))):
        print(''.join(p[x]))
    print('')
```

In [7]:
```
## printing out puzzle twice to see the space after each puzzle
prettyPrint(puzzle)
prettyPrint(puzzle)
```

```
marco
oneup
skate
elder
seeya

marco
oneup
skate
elder
seeya
```

Problem 4.2

Write four calls to `prettyPrint` that slice the given array `puzzle` to result in the following output:

```
mar
one

der
eya

marco

o
p
e
r
a
```

Your code should comply with the given specifications. Show code and output.

Answer

In [8]:

```
## code
prettyPrint(puzzle[:2, :3])
prettyPrint(puzzle[-2:, -3:])
prettyPrint(np.array([puzzle[0, :]]))
prettyPrint(puzzle[:, -1:])
```

```
mar
one

der
eya

marco

o
p
e
r
a
```

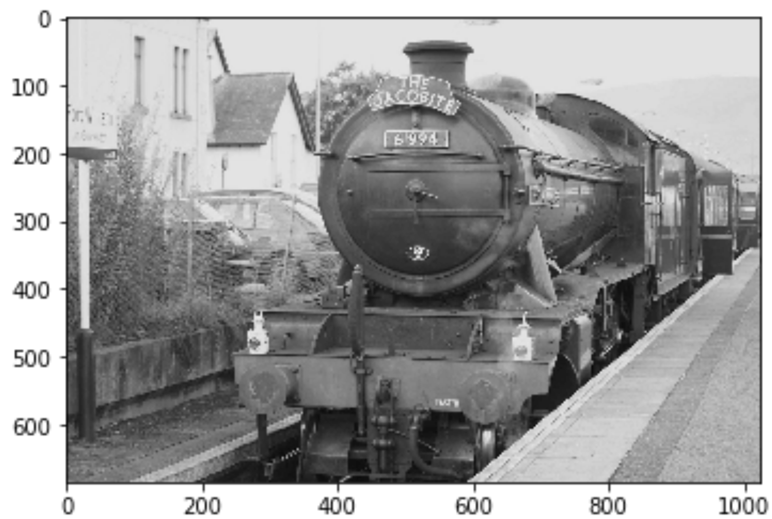
Problem 4.3

In [9]:
```
from skimage import io, color

def readGray(filename):
    img = color.rgb2gray(io.imread(filename))
    return np.around(255 * img).astype(np.uint8)

from matplotlib import pyplot as plt
%matplotlib inline

img = readGray('locomotive.jpg')
plt.imshow(img, cmap='gray')
plt.show()
```



Write a function with header

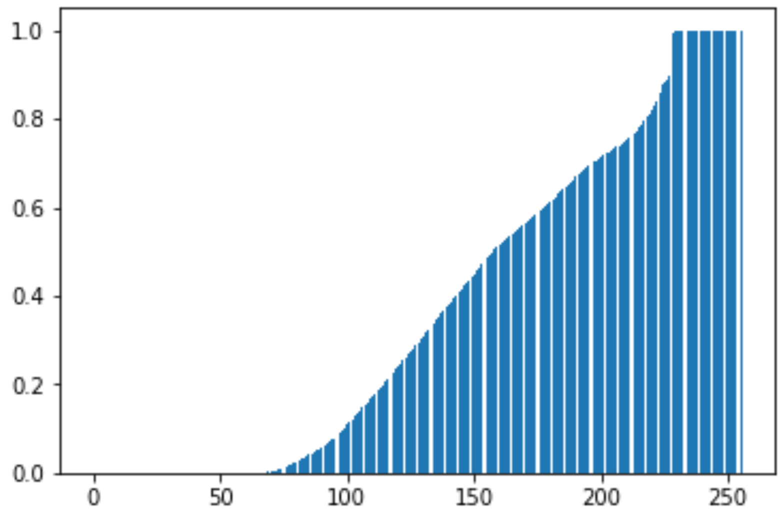
```
def cumulative(img):
```

that takes an image as produced by `readGray` and computes its cumulative distribution $c(k)$. Show your code, and plot the cumulative histogram of the image in `locomotive.jpg` as read by `readGray`.

Answer

In [10]:
```
def cumulative(img):
    ck = {}
    for k in range(256):
        ck[k] = (img < k).sum() / len(img.flatten())
    return ck

ck = cumulative(img)
plt.bar(ck.keys(), ck.values())
plt.show()
```



Problem 4.4

If your cumulative distribution is correct, the plot should start with several values $c(k)$ equal to zero.

What do these zero values mean? In particular, how do these zeros manifest themselves visually in the image?

Answer

These zero values mean that there are no pixels in the image whose value is less than 56. A pixel with a value of zero is *black*. A pixel with a value of 255 is *white*. The several values *c(k)* equal to zero mean that there is no black in the image, and there are no pixels very close in darkness to black in the image either. So, these zeros manifest themselves visually as a *lighter* image.

Problem 4.5

Write a function with header

```
def remap(img):
```

that takes an image as returned by `readGray` and replaces the value k of every pixel with value $k' = \text{round}(255\, c(k))$, where the function $c(k)$ was defined in problem 4.3.

Show your code, display the result of running `remap` on the image in `locomotive.jpg`, and plot the cumulative histogram of the remapped image.

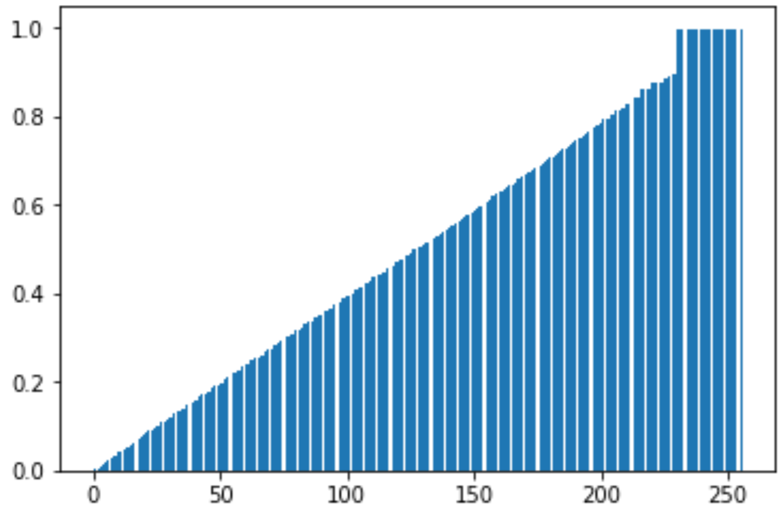
Answer

In [11]:
```
def remap(img):
    ck = cumulative(img)
    imgFlat = img.flatten()
    remapFlat = np.zeros_like(imgFlat)
    for i in range(len(imgFlat)):
        remapFlat[i] = round(255 * ck[imgFlat[i]])
    return remapFlat.reshape(img.shape)

locomotive_remap = remap(readGray('locomotive.jpg'))
locomotive_remap
```

Out[11]:
```
array([[137, 149, 169, ..., 229, 229, 229],
       [ 68,  90, 122, ..., 229, 229, 229],
       [101, 125, 145, ..., 229, 229, 229],
       ...,
       [118, 102, 104, ..., 162, 176, 162],
       [113, 122, 115, ..., 149, 163, 169],
       [109, 116, 113, ..., 151, 156, 148]], dtype=uint8)
```

In [12]:
```
ck_locomotive_remap = cumulative(locomotive_remap)
plt.bar(ck_locomotive_remap.keys(), ck_locomotive_remap.values())
plt.show()
```



Problem 4.6

Comment on the result. In particular, what would the *histogram* (rather than the cumulative distribution) of the remapped image look like, approximately? And in what way does the remapped *image* differ from the original, visually?

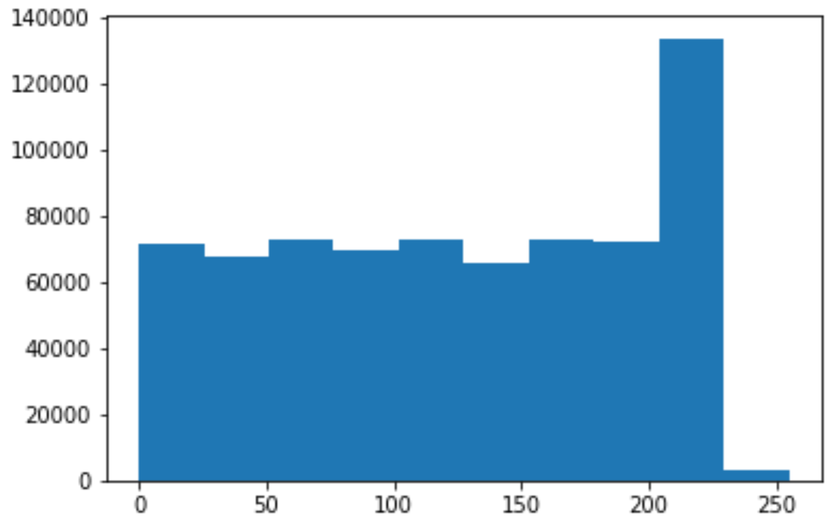
Answer

The *histogram* (rather than the cumulative distribution) of the remapped image looks like, approximately, a uniform distribution.

Visually, the remapped image looks *darker* than the original, with *higher contrast*. Why? Well, because the values of the pixels of the original were all greater than 55. But the remapped image has pixels of lower value, which are *darker*. And the variance of the values of the pixels in the remapped image is higher than that of the original.

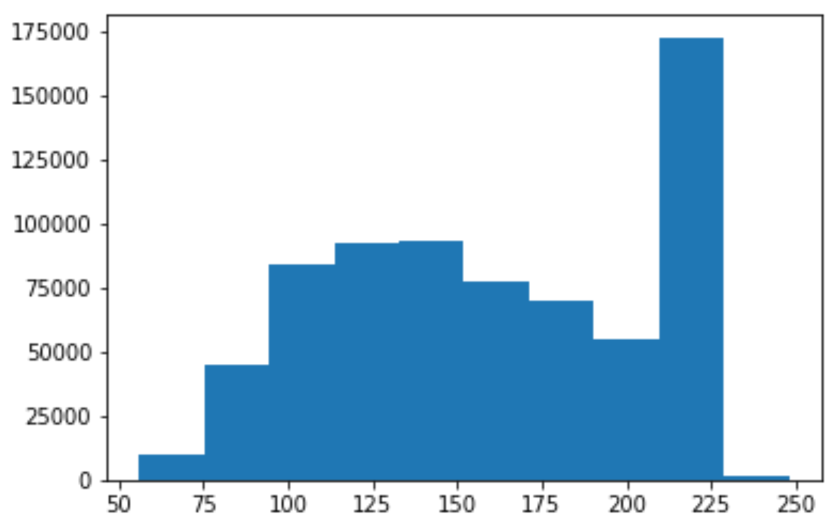
In [13]:

```
# Histogram of remapped image.
plt.hist(locomotive_remap.flatten())
plt.show()
```

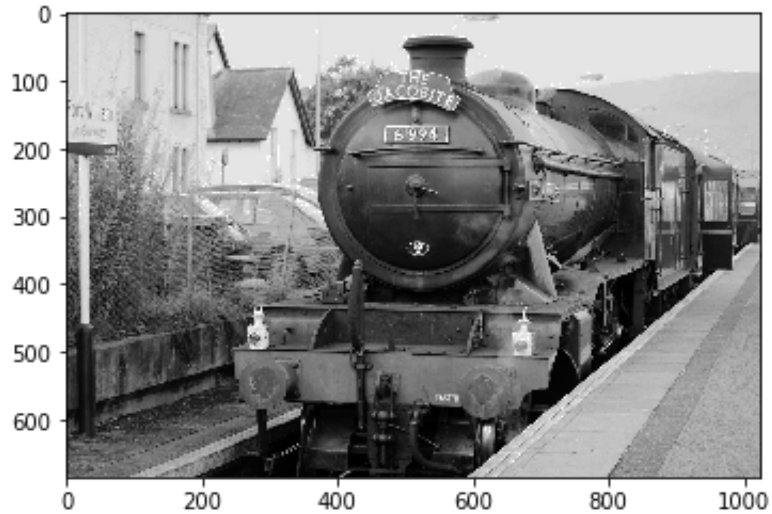


In [14]:

```
# Histogram of original image.
plt.hist(img.flatten())
plt.show()
```



In [15]:
```
# Remapped image.
plt.imshow(locomotive_remap, cmap='gray')
plt.show()
```



In [16]:
```
# Variance of values of pixels of original image.
np.var(img)
```

Out[16]: 2223.0428623519388

In [17]:
```
# Variance of values of pixels of remapped image.
np.var(locomotive_remap)
```

Out[17]: 5151.1616565576105