Name: PRATAP SHINGANE

PRN: 2020BTEIT00050

COURSE: COMPUTER ALGORITHM [LAB]

PS: Fractual Image compression based on principle of self- similarity with steps including compression and decompression domain-range mapping starting with a big range block and subdividing it if an appropriate match is not found.

import matplotlib.pyplot as plt

import matplotlib.image as mpimg

from scipy import ndimage

from scipy import optimize

import numpy as np

import math

# Manipulate channels

def get\_greyscale\_image(img):

    return np.mean(img[:,:,:2], 2)

def extract\_rgb(img):

    return img[:,:,0], img[:,:,1], img[:,:,2]

def assemble\_rbg(img\_r, img\_g, img\_b):

    shape = (img\_r.shape[0], img\_r.shape[1], 1)

    return np.concatenate((np.reshape(img\_r, shape), np.reshape(img\_g, shape),

        np.reshape(img\_b, shape)), axis=2)

# Transformations

def reduce(img, factor):

    result = np.zeros((img.shape[0] // factor, img.shape[1] // factor))

    for i in range(result.shape[0]):

        for j in range(result.shape[1]):

            result[i,j] = np.mean(img[i\*factor:(i+1)\*factor,j\*factor:(j+1)\*factor])

    return result

def rotate(img, angle):

    return ndimage.rotate(img, angle, reshape=False)

def flip(img, direction):

    return img[::direction,:]

def apply\_transformation(img, direction, angle, contrast=1.0, brightness=0.0):

    return contrast\*rotate(flip(img, direction), angle) + brightness

# Contrast and brightness

def find\_contrast\_and\_brightness1(D, S):

    # Fix the contrast and only fit the brightness

    contrast = 0.75

    brightness = (np.sum(D - contrast\*S)) / D.size

    return contrast, brightness

def find\_contrast\_and\_brightness2(D, S):

    # Fit the contrast and the brightness

    A = np.concatenate((np.ones((S.size, 1)), np.reshape(S, (S.size, 1))), axis=1)

    b = np.reshape(D, (D.size,))

    x, \_, \_, \_ = np.linalg.lstsq(A, b)

    #x = optimize.lsq\_linear(A, b, [(-np.inf, -2.0), (np.inf, 2.0)]).x

    return x[1], x[0]

# Compression for greyscale images

def generate\_all\_transformed\_blocks(img, source\_size, destination\_size, step):

    factor = source\_size // destination\_size

    transformed\_blocks = []

    for k in range((img.shape[0] - source\_size) // step + 1):

        for l in range((img.shape[1] - source\_size) // step + 1):

            # Extract the source block and reduce it to the shape of a destination block

            S = reduce(img[k\*step:k\*step+source\_size,l\*step:l\*step+source\_size], factor)

            # Generate all possible transformed blocks

            for direction, angle in candidates:

                transformed\_blocks.append((k, l, direction, angle, apply\_transformation(S, direction, angle)))

    return transformed\_blocks

def compress(img, source\_size, destination\_size, step):

    transformations = []

    transformed\_blocks = generate\_all\_transformed\_blocks(img, source\_size, destination\_size, step)

    i\_count = img.shape[0] // destination\_size

    j\_count = img.shape[1] // destination\_size

    for i in range(i\_count):

        transformations.append([])

        for j in range(j\_count):

            print("{}/{} ; {}/{}".format(i, i\_count, j, j\_count))

            transformations[i].append(None)

            min\_d = float('inf')

            # Extract the destination block

            D = img[i\*destination\_size:(i+1)\*destination\_size,j\*destination\_size:(j+1)\*destination\_size]

            # Test all possible transformations and take the best one

            for k, l, direction, angle, S in transformed\_blocks:

                contrast, brightness = find\_contrast\_and\_brightness2(D, S)

                S = contrast\*S + brightness

                d = np.sum(np.square(D - S))

                if d < min\_d:

                    min\_d = d

                    transformations[i][j] = (k, l, direction, angle, contrast, brightness)

    return transformations

def decompress(transformations, source\_size, destination\_size, step, nb\_iter=8):

    factor = source\_size // destination\_size

    height = len(transformations) \* destination\_size

    width = len(transformations[0]) \* destination\_size

    iterations = [np.random.randint(0, 256, (height, width))]

    cur\_img = np.zeros((height, width))

    for i\_iter in range(nb\_iter):

        print(i\_iter)

        for i in range(len(transformations)):

            for j in range(len(transformations[i])):

                # Apply transform

                k, l, flip, angle, contrast, brightness = transformations[i][j]

                S = reduce(iterations[-1][k\*step:k\*step+source\_size,l\*step:l\*step+source\_size], factor)

                D = apply\_transformation(S, flip, angle, contrast, brightness)

                cur\_img[i\*destination\_size:(i+1)\*destination\_size,j\*destination\_size:(j+1)\*destination\_size] = D

        iterations.append(cur\_img)

        cur\_img = np.zeros((height, width))

    return iterations

# Compression for color images

def reduce\_rgb(img, factor):

    img\_r, img\_g, img\_b = extract\_rgb(img)

    img\_r = reduce(img\_r, factor)

    img\_g = reduce(img\_g, factor)

    img\_b = reduce(img\_b, factor)

    return assemble\_rbg(img\_r, img\_g, img\_b)

def compress\_rgb(img, source\_size, destination\_size, step):

    img\_r, img\_g, img\_b = extract\_rgb(img)

    return [compress(img\_r, source\_size, destination\_size, step), \

        compress(img\_g, source\_size, destination\_size, step), \

        compress(img\_b, source\_size, destination\_size, step)]

def decompress\_rgb(transformations, source\_size, destination\_size, step, nb\_iter=8):

    img\_r = decompress(transformations[0], source\_size, destination\_size, step, nb\_iter)[-1]

    img\_g = decompress(transformations[1], source\_size, destination\_size, step, nb\_iter)[-1]

    img\_b = decompress(transformations[2], source\_size, destination\_size, step, nb\_iter)[-1]

    return assemble\_rbg(img\_r, img\_g, img\_b)

# Plot

def plot\_iterations(iterations, target=None):

    # Configure plot

    plt.figure()

    nb\_row = math.ceil(np.sqrt(len(iterations)))

    nb\_cols = nb\_row

    # Plot

    for i, img in enumerate(iterations):

        plt.subplot(nb\_row, nb\_cols, i+1)

        plt.imshow(img, cmap='gray', vmin=0, vmax=255, interpolation='none')

        if target is None:

            plt.title(str(i))

        else:

            # Display the RMSE

            plt.title(str(i) + ' (' + '{0:.2f}'.format(np.sqrt(np.mean(np.square(target - img)))) + ')')

        frame = plt.gca()

        frame.axes.get\_xaxis().set\_visible(False)

        frame.axes.get\_yaxis().set\_visible(False)

    plt.tight\_layout()

# Parameters

directions = [1, -1]

angles = [0, 90, 180, 270]

candidates = [[direction, angle] for direction in directions for angle in angles]

# Tests

def test\_greyscale():

    img = mpimg.imread('/content/sample\_data/lena.gif')

    img = get\_greyscale\_image(img)

    img = reduce(img, 4)

    plt.figure()

    plt.imshow(img, cmap='gray', interpolation='none')

    transformations = compress(img, 8, 4, 8)

    iterations = decompress(transformations, 8, 4, 8)

    plot\_iterations(iterations, img)

    plt.show()

def test\_rgb():

    img = mpimg.imread('/content/sample\_data/monkey.gif')

    img = reduce\_rgb(img, 8)

    transformations = compress\_rgb(img, 8, 4, 8)

    retrieved\_img = decompress\_rgb(transformations, 8, 4, 8)

    plt.figure()

    plt.subplot(121)

    plt.imshow(np.array(img).astype(np.uint8), interpolation='none')

    plt.subplot(122)

    plt.imshow(retrieved\_img.astype(np.uint8), interpolation='none')

    plt.show()

if \_\_name\_\_ == '\_\_main\_\_':

    test\_greyscale()

    #test\_rgb()

Output:

0/32 ; 0/32

0/32 ; 1/32

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:51: FutureWarning: `rcond` parameter will change to the default of machine precision times ``max(M, N)`` where M and N are the input matrix dimensions.

To use the future default and silence this warning we advise to pass `rcond=None`, to keep using the old, explicitly pass `rcond=-1`.

0/32 ; 2/32

0/32 ; 3/32

0/32 ; 4/32

0/32 ; 5/32

0/32 ; 6/32

0/32 ; 7/32

0/32 ; 8/32

0/32 ; 9/32

0/32 ; 10/32

0/32 ; 11/32

0/32 ; 12/32

0/32 ; 13/32

0/32 ; 14/32

0/32 ; 15/32

0/32 ; 16/32

0/32 ; 17/32

0/32 ; 18/32

0/32 ; 19/32

0/32 ; 20/32

0/32 ; 21/32

0/32 ; 22/32

0/32 ; 23/32

0/32 ; 24/32

0/32 ; 25/32

0/32 ; 26/32

0/32 ; 27/32

0/32 ; 28/32

0/32 ; 29/32

0/32 ; 30/32

0/32 ; 31/32

1/32 ; 0/32

1/32 ; 1/32

1/32 ; 2/32

1/32 ; 3/32

1/32 ; 4/32

1/32 ; 5/32

1/32 ; 6/32

1/32 ; 7/32

1/32 ; 8/32

1/32 ; 9/32

1/32 ; 10/32

1/32 ; 11/32

1/32 ; 12/32

1/32 ; 13/32

1/32 ; 14/32

1/32 ; 15/32

1/32 ; 16/32

1/32 ; 17/32

1/32 ; 18/32

1/32 ; 19/32

1/32 ; 20/32

1/32 ; 21/32

1/32 ; 22/32

1/32 ; 23/32

1/32 ; 24/32

1/32 ; 25/32

1/32 ; 26/32

1/32 ; 27/32

1/32 ; 28/32

1/32 ; 29/32

1/32 ; 30/32

1/32 ; 31/32

2/32 ; 0/32

2/32 ; 1/32

2/32 ; 2/32

2/32 ; 3/32

2/32 ; 4/32

2/32 ; 5/32

2/32 ; 6/32

2/32 ; 7/32

2/32 ; 8/32

2/32 ; 9/32

2/32 ; 10/32

2/32 ; 11/32

2/32 ; 12/32

2/32 ; 13/32

2/32 ; 14/32

2/32 ; 15/32

2/32 ; 16/32

2/32 ; 17/32

2/32 ; 18/32

2/32 ; 19/32

2/32 ; 20/32

2/32 ; 21/32

2/32 ; 22/32

2/32 ; 23/32

2/32 ; 24/32

2/32 ; 25/32

2/32 ; 26/32

2/32 ; 27/32

2/32 ; 28/32

2/32 ; 29/32

2/32 ; 30/32

2/32 ; 31/32

3/32 ; 0/32

3/32 ; 1/32

3/32 ; 2/32

3/32 ; 3/32

3/32 ; 4/32

3/32 ; 5/32

3/32 ; 6/32

3/32 ; 7/32

3/32 ; 8/32

3/32 ; 9/32

3/32 ; 10/32

3/32 ; 11/32

3/32 ; 12/32

3/32 ; 13/32

3/32 ; 14/32

3/32 ; 15/32

3/32 ; 16/32

3/32 ; 17/32

3/32 ; 18/32

3/32 ; 19/32

3/32 ; 20/32

3/32 ; 21/32

3/32 ; 22/32

3/32 ; 23/32

3/32 ; 24/32

3/32 ; 25/32

3/32 ; 26/32

3/32 ; 27/32

3/32 ; 28/32

3/32 ; 29/32

3/32 ; 30/32

3/32 ; 31/32

4/32 ; 0/32

4/32 ; 1/32

4/32 ; 2/32

4/32 ; 3/32

4/32 ; 4/32

4/32 ; 5/32

4/32 ; 6/32

4/32 ; 7/32

4/32 ; 8/32

4/32 ; 9/32

4/32 ; 10/32

4/32 ; 11/32

4/32 ; 12/32

4/32 ; 13/32

4/32 ; 14/32

4/32 ; 15/32

4/32 ; 16/32

4/32 ; 17/32

4/32 ; 18/32

4/32 ; 19/32

4/32 ; 20/32

4/32 ; 21/32

4/32 ; 22/32

4/32 ; 23/32

4/32 ; 24/32

4/32 ; 25/32

4/32 ; 26/32

4/32 ; 27/32

4/32 ; 28/32

4/32 ; 29/32

4/32 ; 30/32

4/32 ; 31/32

5/32 ; 0/32

5/32 ; 1/32

5/32 ; 2/32

5/32 ; 3/32

5/32 ; 4/32

5/32 ; 5/32

5/32 ; 6/32

5/32 ; 7/32

5/32 ; 8/32

5/32 ; 9/32

5/32 ; 10/32

5/32 ; 11/32

5/32 ; 12/32

5/32 ; 13/32

5/32 ; 14/32

5/32 ; 15/32

5/32 ; 16/32

5/32 ; 17/32

5/32 ; 18/32

5/32 ; 19/32

5/32 ; 20/32

5/32 ; 21/32

5/32 ; 22/32

5/32 ; 23/32

5/32 ; 24/32

5/32 ; 25/32

5/32 ; 26/32

5/32 ; 27/32

5/32 ; 28/32

5/32 ; 29/32

5/32 ; 30/32

5/32 ; 31/32

6/32 ; 0/32

6/32 ; 1/32

6/32 ; 2/32

6/32 ; 3/32

6/32 ; 4/32

6/32 ; 5/32

6/32 ; 6/32

6/32 ; 7/32

6/32 ; 8/32

6/32 ; 9/32

6/32 ; 10/32

6/32 ; 11/32

6/32 ; 12/32

6/32 ; 13/32

6/32 ; 14/32

6/32 ; 15/32

6/32 ; 16/32

6/32 ; 17/32

6/32 ; 18/32

6/32 ; 19/32

6/32 ; 20/32

6/32 ; 21/32

6/32 ; 22/32

6/32 ; 23/32

6/32 ; 24/32

6/32 ; 25/32

6/32 ; 26/32

6/32 ; 27/32

6/32 ; 28/32

6/32 ; 29/32

6/32 ; 30/32

6/32 ; 31/32

7/32 ; 0/32

7/32 ; 1/32

7/32 ; 2/32

7/32 ; 3/32

7/32 ; 4/32

7/32 ; 5/32

7/32 ; 6/32

7/32 ; 7/32

7/32 ; 8/32

7/32 ; 9/32

7/32 ; 10/32

7/32 ; 11/32

7/32 ; 12/32

7/32 ; 13/32

7/32 ; 14/32

7/32 ; 15/32

7/32 ; 16/32

7/32 ; 17/32

7/32 ; 18/32

7/32 ; 19/32

7/32 ; 20/32

7/32 ; 21/32

7/32 ; 22/32

7/32 ; 23/32

7/32 ; 24/32

7/32 ; 25/32

7/32 ; 26/32

7/32 ; 27/32

7/32 ; 28/32

7/32 ; 29/32

7/32 ; 30/32

7/32 ; 31/32

8/32 ; 0/32

8/32 ; 1/32

8/32 ; 2/32

8/32 ; 3/32

8/32 ; 4/32

8/32 ; 5/32

8/32 ; 6/32

8/32 ; 7/32

8/32 ; 8/32

8/32 ; 9/32

8/32 ; 10/32

8/32 ; 11/32

8/32 ; 12/32

8/32 ; 13/32

8/32 ; 14/32

8/32 ; 15/32

8/32 ; 16/32

8/32 ; 17/32

8/32 ; 18/32

8/32 ; 19/32

8/32 ; 20/32

8/32 ; 21/32

8/32 ; 22/32

8/32 ; 23/32

8/32 ; 24/32

8/32 ; 25/32

8/32 ; 26/32

8/32 ; 27/32

8/32 ; 28/32

8/32 ; 29/32

8/32 ; 30/32

8/32 ; 31/32

9/32 ; 0/32

9/32 ; 1/32

9/32 ; 2/32

9/32 ; 3/32

9/32 ; 4/32

9/32 ; 5/32

9/32 ; 6/32

9/32 ; 7/32

9/32 ; 8/32

9/32 ; 9/32

9/32 ; 10/32

9/32 ; 11/32

9/32 ; 12/32

9/32 ; 13/32

9/32 ; 14/32

9/32 ; 15/32

9/32 ; 16/32

9/32 ; 17/32

9/32 ; 18/32

9/32 ; 19/32

9/32 ; 20/32

9/32 ; 21/32

9/32 ; 22/32

9/32 ; 23/32

9/32 ; 24/32

9/32 ; 25/32

9/32 ; 26/32

9/32 ; 27/32

9/32 ; 28/32

9/32 ; 29/32

9/32 ; 30/32

9/32 ; 31/32

10/32 ; 0/32

10/32 ; 1/32

10/32 ; 2/32

10/32 ; 3/32

10/32 ; 4/32

10/32 ; 5/32

10/32 ; 6/32

10/32 ; 7/32

10/32 ; 8/32

10/32 ; 9/32

10/32 ; 10/32

10/32 ; 11/32

10/32 ; 12/32

10/32 ; 13/32

10/32 ; 14/32

10/32 ; 15/32

10/32 ; 16/32

10/32 ; 17/32

10/32 ; 18/32

10/32 ; 19/32

10/32 ; 20/32

10/32 ; 21/32

10/32 ; 22/32

10/32 ; 23/32

10/32 ; 24/32

10/32 ; 25/32

10/32 ; 26/32

10/32 ; 27/32

10/32 ; 28/32

10/32 ; 29/32

10/32 ; 30/32

10/32 ; 31/32

11/32 ; 0/32

11/32 ; 1/32

11/32 ; 2/32

11/32 ; 3/32

11/32 ; 4/32

11/32 ; 5/32

11/32 ; 6/32

11/32 ; 7/32

11/32 ; 8/32

11/32 ; 9/32

11/32 ; 10/32

11/32 ; 11/32

11/32 ; 12/32

11/32 ; 13/32

11/32 ; 14/32

11/32 ; 15/32

11/32 ; 16/32

11/32 ; 17/32

11/32 ; 18/32

11/32 ; 19/32

11/32 ; 20/32

11/32 ; 21/32

11/32 ; 22/32

11/32 ; 23/32

11/32 ; 24/32

11/32 ; 25/32

11/32 ; 26/32

11/32 ; 27/32

11/32 ; 28/32

11/32 ; 29/32

11/32 ; 30/32

11/32 ; 31/32

12/32 ; 0/32

12/32 ; 1/32

12/32 ; 2/32

12/32 ; 3/32

12/32 ; 4/32

12/32 ; 5/32

12/32 ; 6/32

12/32 ; 7/32

12/32 ; 8/32

12/32 ; 9/32

12/32 ; 10/32

12/32 ; 11/32

12/32 ; 12/32

12/32 ; 13/32

12/32 ; 14/32

12/32 ; 15/32

12/32 ; 16/32

12/32 ; 17/32

12/32 ; 18/32

12/32 ; 19/32

12/32 ; 20/32

12/32 ; 21/32

12/32 ; 22/32

12/32 ; 23/32

12/32 ; 24/32

12/32 ; 25/32

12/32 ; 26/32

12/32 ; 27/32

12/32 ; 28/32

12/32 ; 29/32

12/32 ; 30/32

12/32 ; 31/32

13/32 ; 0/32

13/32 ; 1/32

13/32 ; 2/32

13/32 ; 3/32

13/32 ; 4/32

13/32 ; 5/32

13/32 ; 6/32

13/32 ; 7/32

13/32 ; 8/32

13/32 ; 9/32

13/32 ; 10/32

13/32 ; 11/32

13/32 ; 12/32

13/32 ; 13/32

13/32 ; 14/32

13/32 ; 15/32

13/32 ; 16/32

13/32 ; 17/32

13/32 ; 18/32

13/32 ; 19/32

13/32 ; 20/32

13/32 ; 21/32

13/32 ; 22/32

13/32 ; 23/32

13/32 ; 24/32

13/32 ; 25/32

13/32 ; 26/32

13/32 ; 27/32

13/32 ; 28/32

13/32 ; 29/32

13/32 ; 30/32

13/32 ; 31/32

14/32 ; 0/32

14/32 ; 1/32

14/32 ; 2/32

14/32 ; 3/32

14/32 ; 4/32

14/32 ; 5/32

14/32 ; 6/32

14/32 ; 7/32

14/32 ; 8/32

14/32 ; 9/32

14/32 ; 10/32

14/32 ; 11/32

14/32 ; 12/32

14/32 ; 13/32

14/32 ; 14/32

14/32 ; 15/32

14/32 ; 16/32

14/32 ; 17/32

14/32 ; 18/32

14/32 ; 19/32

14/32 ; 20/32

14/32 ; 21/32

14/32 ; 22/32

14/32 ; 23/32

14/32 ; 24/32

14/32 ; 25/32

14/32 ; 26/32

14/32 ; 27/32

14/32 ; 28/32

14/32 ; 29/32

14/32 ; 30/32

14/32 ; 31/32

15/32 ; 0/32

15/32 ; 1/32

15/32 ; 2/32

15/32 ; 3/32

15/32 ; 4/32

15/32 ; 5/32

15/32 ; 6/32

15/32 ; 7/32

15/32 ; 8/32

15/32 ; 9/32

15/32 ; 10/32

15/32 ; 11/32

15/32 ; 12/32

15/32 ; 13/32

15/32 ; 14/32

15/32 ; 15/32

15/32 ; 16/32

15/32 ; 17/32

15/32 ; 18/32

15/32 ; 19/32

15/32 ; 20/32

15/32 ; 21/32

15/32 ; 22/32

15/32 ; 23/32

15/32 ; 24/32

15/32 ; 25/32

15/32 ; 26/32

15/32 ; 27/32

15/32 ; 28/32

15/32 ; 29/32

15/32 ; 30/32

15/32 ; 31/32

16/32 ; 0/32

16/32 ; 1/32

16/32 ; 2/32

16/32 ; 3/32

16/32 ; 4/32

16/32 ; 5/32

16/32 ; 6/32

16/32 ; 7/32

16/32 ; 8/32

16/32 ; 9/32

16/32 ; 10/32

16/32 ; 11/32

16/32 ; 12/32

16/32 ; 13/32

16/32 ; 14/32

16/32 ; 15/32

16/32 ; 16/32

16/32 ; 17/32

16/32 ; 18/32

16/32 ; 19/32

16/32 ; 20/32

16/32 ; 21/32

16/32 ; 22/32

16/32 ; 23/32

16/32 ; 24/32

16/32 ; 25/32

16/32 ; 26/32

16/32 ; 27/32

16/32 ; 28/32

16/32 ; 29/32

16/32 ; 30/32

16/32 ; 31/32

17/32 ; 0/32

17/32 ; 1/32

17/32 ; 2/32

17/32 ; 3/32

17/32 ; 4/32

17/32 ; 5/32

17/32 ; 6/32

17/32 ; 7/32

17/32 ; 8/32

17/32 ; 9/32

17/32 ; 10/32

17/32 ; 11/32

17/32 ; 12/32

17/32 ; 13/32

17/32 ; 14/32

17/32 ; 15/32

17/32 ; 16/32

17/32 ; 17/32

17/32 ; 18/32

17/32 ; 19/32

17/32 ; 20/32

17/32 ; 21/32

17/32 ; 22/32

17/32 ; 23/32

17/32 ; 24/32

17/32 ; 25/32

17/32 ; 26/32

17/32 ; 27/32

17/32 ; 28/32

17/32 ; 29/32

17/32 ; 30/32

17/32 ; 31/32

18/32 ; 0/32

18/32 ; 1/32

18/32 ; 2/32

18/32 ; 3/32

18/32 ; 4/32

18/32 ; 5/32

18/32 ; 6/32

18/32 ; 7/32

18/32 ; 8/32

18/32 ; 9/32

18/32 ; 10/32

18/32 ; 11/32

18/32 ; 12/32

18/32 ; 13/32

18/32 ; 14/32

18/32 ; 15/32

18/32 ; 16/32

18/32 ; 17/32

18/32 ; 18/32

18/32 ; 19/32

18/32 ; 20/32

18/32 ; 21/32

18/32 ; 22/32

18/32 ; 23/32

18/32 ; 24/32

18/32 ; 25/32

18/32 ; 26/32

18/32 ; 27/32

18/32 ; 28/32

18/32 ; 29/32

18/32 ; 30/32

18/32 ; 31/32

19/32 ; 0/32

19/32 ; 1/32

19/32 ; 2/32

19/32 ; 3/32

19/32 ; 4/32

19/32 ; 5/32

19/32 ; 6/32

19/32 ; 7/32

19/32 ; 8/32

19/32 ; 9/32

19/32 ; 10/32

19/32 ; 11/32

19/32 ; 12/32

19/32 ; 13/32

19/32 ; 14/32

19/32 ; 15/32

19/32 ; 16/32

19/32 ; 17/32

19/32 ; 18/32

19/32 ; 19/32

19/32 ; 20/32

19/32 ; 21/32

19/32 ; 22/32

19/32 ; 23/32

19/32 ; 24/32

19/32 ; 25/32

19/32 ; 26/32

19/32 ; 27/32

19/32 ; 28/32

19/32 ; 29/32

19/32 ; 30/32

19/32 ; 31/32

20/32 ; 0/32

20/32 ; 1/32

20/32 ; 2/32

20/32 ; 3/32

20/32 ; 4/32

20/32 ; 5/32

20/32 ; 6/32

20/32 ; 7/32

20/32 ; 8/32

20/32 ; 9/32

20/32 ; 10/32

20/32 ; 11/32

20/32 ; 12/32

20/32 ; 13/32

20/32 ; 14/32

20/32 ; 15/32

20/32 ; 16/32

20/32 ; 17/32

20/32 ; 18/32

20/32 ; 19/32

20/32 ; 20/32

20/32 ; 21/32

20/32 ; 22/32

20/32 ; 23/32

20/32 ; 24/32

20/32 ; 25/32

20/32 ; 26/32

20/32 ; 27/32

20/32 ; 28/32

20/32 ; 29/32

20/32 ; 30/32

20/32 ; 31/32

21/32 ; 0/32

21/32 ; 1/32

21/32 ; 2/32

21/32 ; 3/32

21/32 ; 4/32

21/32 ; 5/32

21/32 ; 6/32

21/32 ; 7/32

21/32 ; 8/32

21/32 ; 9/32

21/32 ; 10/32

21/32 ; 11/32

21/32 ; 12/32

21/32 ; 13/32

21/32 ; 14/32

21/32 ; 15/32

21/32 ; 16/32

21/32 ; 17/32

21/32 ; 18/32

21/32 ; 19/32

21/32 ; 20/32

21/32 ; 21/32

21/32 ; 22/32

21/32 ; 23/32

21/32 ; 24/32

21/32 ; 25/32

21/32 ; 26/32

21/32 ; 27/32

21/32 ; 28/32

21/32 ; 29/32

21/32 ; 30/32

21/32 ; 31/32

22/32 ; 0/32

22/32 ; 1/32

22/32 ; 2/32

22/32 ; 3/32

22/32 ; 4/32

22/32 ; 5/32

22/32 ; 6/32

22/32 ; 7/32

22/32 ; 8/32

22/32 ; 9/32

22/32 ; 10/32

22/32 ; 11/32

22/32 ; 12/32

22/32 ; 13/32

22/32 ; 14/32

22/32 ; 15/32

22/32 ; 16/32

22/32 ; 17/32

22/32 ; 18/32

22/32 ; 19/32

22/32 ; 20/32

22/32 ; 21/32

22/32 ; 22/32

22/32 ; 23/32

22/32 ; 24/32

22/32 ; 25/32

22/32 ; 26/32

22/32 ; 27/32

22/32 ; 28/32

22/32 ; 29/32

22/32 ; 30/32

22/32 ; 31/32

23/32 ; 0/32

23/32 ; 1/32

23/32 ; 2/32

23/32 ; 3/32

23/32 ; 4/32

23/32 ; 5/32

23/32 ; 6/32

23/32 ; 7/32

23/32 ; 8/32

23/32 ; 9/32

23/32 ; 10/32

23/32 ; 11/32

23/32 ; 12/32

23/32 ; 13/32

23/32 ; 14/32

23/32 ; 15/32

23/32 ; 16/32

23/32 ; 17/32

23/32 ; 18/32

23/32 ; 19/32

23/32 ; 20/32

23/32 ; 21/32

23/32 ; 22/32

23/32 ; 23/32

23/32 ; 24/32

23/32 ; 25/32

23/32 ; 26/32

23/32 ; 27/32

23/32 ; 28/32

23/32 ; 29/32

23/32 ; 30/32

23/32 ; 31/32

24/32 ; 0/32

24/32 ; 1/32

24/32 ; 2/32

24/32 ; 3/32

24/32 ; 4/32

24/32 ; 5/32

24/32 ; 6/32

24/32 ; 7/32

24/32 ; 8/32

24/32 ; 9/32

24/32 ; 10/32

24/32 ; 11/32

24/32 ; 12/32

24/32 ; 13/32

24/32 ; 14/32

24/32 ; 15/32

24/32 ; 16/32

24/32 ; 17/32

24/32 ; 18/32

24/32 ; 19/32

24/32 ; 20/32

24/32 ; 21/32

24/32 ; 22/32

24/32 ; 23/32

24/32 ; 24/32

24/32 ; 25/32

24/32 ; 26/32

24/32 ; 27/32

24/32 ; 28/32

24/32 ; 29/32

24/32 ; 30/32

24/32 ; 31/32

25/32 ; 0/32

25/32 ; 1/32

25/32 ; 2/32

25/32 ; 3/32

25/32 ; 4/32

25/32 ; 5/32

25/32 ; 6/32

25/32 ; 7/32

25/32 ; 8/32

25/32 ; 9/32

25/32 ; 10/32

25/32 ; 11/32

25/32 ; 12/32

25/32 ; 13/32

25/32 ; 14/32

25/32 ; 15/32

25/32 ; 16/32

25/32 ; 17/32

25/32 ; 18/32

25/32 ; 19/32

25/32 ; 20/32

25/32 ; 21/32

25/32 ; 22/32

25/32 ; 23/32

25/32 ; 24/32

25/32 ; 25/32

25/32 ; 26/32

25/32 ; 27/32

25/32 ; 28/32

25/32 ; 29/32

25/32 ; 30/32

25/32 ; 31/32

26/32 ; 0/32

26/32 ; 1/32

26/32 ; 2/32

26/32 ; 3/32

26/32 ; 4/32

26/32 ; 5/32

26/32 ; 6/32

26/32 ; 7/32

26/32 ; 8/32

26/32 ; 9/32

26/32 ; 10/32

26/32 ; 11/32

26/32 ; 12/32

26/32 ; 13/32

26/32 ; 14/32

26/32 ; 15/32

26/32 ; 16/32

26/32 ; 17/32

26/32 ; 18/32

26/32 ; 19/32

26/32 ; 20/32

26/32 ; 21/32

26/32 ; 22/32

26/32 ; 23/32

26/32 ; 24/32

26/32 ; 25/32

26/32 ; 26/32

26/32 ; 27/32

26/32 ; 28/32

26/32 ; 29/32

26/32 ; 30/32

26/32 ; 31/32

27/32 ; 0/32

27/32 ; 1/32

27/32 ; 2/32

27/32 ; 3/32

27/32 ; 4/32

27/32 ; 5/32

27/32 ; 6/32

27/32 ; 7/32

27/32 ; 8/32

27/32 ; 9/32

27/32 ; 10/32

27/32 ; 11/32

27/32 ; 12/32

27/32 ; 13/32

27/32 ; 14/32

27/32 ; 15/32

27/32 ; 16/32

27/32 ; 17/32

27/32 ; 18/32

27/32 ; 19/32

27/32 ; 20/32

27/32 ; 21/32

27/32 ; 22/32

27/32 ; 23/32

27/32 ; 24/32

27/32 ; 25/32

27/32 ; 26/32

27/32 ; 27/32

27/32 ; 28/32

27/32 ; 29/32

27/32 ; 30/32

27/32 ; 31/32

28/32 ; 0/32

28/32 ; 1/32

28/32 ; 2/32

28/32 ; 3/32

28/32 ; 4/32

28/32 ; 5/32

28/32 ; 6/32

28/32 ; 7/32

28/32 ; 8/32

28/32 ; 9/32

28/32 ; 10/32

28/32 ; 11/32

28/32 ; 12/32

28/32 ; 13/32

28/32 ; 14/32

28/32 ; 15/32

28/32 ; 16/32

28/32 ; 17/32

28/32 ; 18/32

28/32 ; 19/32

28/32 ; 20/32

28/32 ; 21/32

28/32 ; 22/32

28/32 ; 23/32

28/32 ; 24/32

28/32 ; 25/32

28/32 ; 26/32

28/32 ; 27/32

28/32 ; 28/32

28/32 ; 29/32

28/32 ; 30/32

28/32 ; 31/32

29/32 ; 0/32

29/32 ; 1/32

29/32 ; 2/32

29/32 ; 3/32

29/32 ; 4/32

29/32 ; 5/32

29/32 ; 6/32

29/32 ; 7/32

29/32 ; 8/32

29/32 ; 9/32

29/32 ; 10/32

29/32 ; 11/32

29/32 ; 12/32

29/32 ; 13/32

29/32 ; 14/32

29/32 ; 15/32

29/32 ; 16/32

29/32 ; 17/32

29/32 ; 18/32

29/32 ; 19/32

29/32 ; 20/32

29/32 ; 21/32

29/32 ; 22/32

29/32 ; 23/32

29/32 ; 24/32

29/32 ; 25/32

29/32 ; 26/32

29/32 ; 27/32

29/32 ; 28/32

29/32 ; 29/32

29/32 ; 30/32

29/32 ; 31/32

30/32 ; 0/32

30/32 ; 1/32

30/32 ; 2/32

30/32 ; 3/32

30/32 ; 4/32

30/32 ; 5/32

30/32 ; 6/32

30/32 ; 7/32

30/32 ; 8/32

30/32 ; 9/32

30/32 ; 10/32

30/32 ; 11/32

30/32 ; 12/32

30/32 ; 13/32

30/32 ; 14/32

30/32 ; 15/32

30/32 ; 16/32

30/32 ; 17/32

30/32 ; 18/32

30/32 ; 19/32

30/32 ; 20/32

30/32 ; 21/32

30/32 ; 22/32

30/32 ; 23/32

30/32 ; 24/32

30/32 ; 25/32

30/32 ; 26/32

30/32 ; 27/32

30/32 ; 28/32

30/32 ; 29/32

30/32 ; 30/32

30/32 ; 31/32

31/32 ; 0/32

31/32 ; 1/32

31/32 ; 2/32

31/32 ; 3/32

31/32 ; 4/32

31/32 ; 5/32

31/32 ; 6/32

31/32 ; 7/32

31/32 ; 8/32

31/32 ; 9/32

31/32 ; 10/32

31/32 ; 11/32

31/32 ; 12/32

31/32 ; 13/32

31/32 ; 14/32

31/32 ; 15/32

31/32 ; 16/32

31/32 ; 17/32

31/32 ; 18/32

31/32 ; 19/32

31/32 ; 20/32

31/32 ; 21/32

31/32 ; 22/32

31/32 ; 23/32

31/32 ; 24/32

31/32 ; 25/32

31/32 ; 26/32

31/32 ; 27/32

31/32 ; 28/32

31/32 ; 29/32

31/32 ; 30/32

31/32 ; 31/32

0

1

2

3

4

5

6

7



