Power Function

Power law transformations have the following form

$S = C r ^ {\gamma}$

S = output pixel

r = input pixel

C = constant

Map a narrow range of dark input values into a wider range of output values or vice versa Varying $\gamma$ gives a whole family of curves

In [1]:

```
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt
```

In [2]:

```
def power_function(input_image):
  img = input_image.resize((400,400), Image.Resampling.LANCZOS)



  # convert to numpy array
  numpy_image = np.array(img)
  numpy_image = numpy_image/255

  #transforming for different values of gamma

  #plotting input and output images
  # set up side-by-side image display
  fig = plt.figure()
  fig.set_figheight(20)
  fig.set_figwidth(20)

  fig.add_subplot(4,4,1)
  plt.imshow(img, cmap='gray')
  plt.title('original image')

  gamma = 0.1
  for i in range(15):
    #print(i*gamma)
    numpy_image_a = np.power(numpy_image,gamma*(i+1))
    numpy_image_a = numpy_image_a * 255
    numpy_image_a = np.around(numpy_image_a,decimals=0)
    power_image = Image.fromarray(numpy_image_a)
    power_image = power_image.convert("L")
    fig.add_subplot(4,4,i+2)
    plt.imshow(power_image, cmap='gray')
    plt.title('gamma = '+ str(round(i*gamma,1)))


  return power_image
```

In [3]:

```
# reading image and converting to gray scale
img = Image.open('../images/tiger.jpg').convert('L')
# Calling the power function
a = power_function(img)
```