PPM P3 Image Reader

Given a PPM (Portable Pixmap) file in P3 format (ASCII), read the image data and return it as a 3-dimensional List. PPM P3 files store RGB color images in a human-readable ASCII format.

Your task is to parse the PPM P3 file and return the image data as a List with shape (H, W, C) where H is height, W is width, and C is the number of color channels (3 for RGB).

Input Format

A filename string representing the path to a PPM P3 format image file.

Output Format

A 3d **list** with shape (W, H, 3) where:

- W: Width of the image in pixels (integer)
- H: Height of the image in pixels (integer)
- 3: Number of color channels (Red, Green, Blue)

The list should contain integers in the range [0, 255] representing the RGB color values.

Constraints

- The input file is a valid PPM P3 format file without any comments
- The maximum color value is 255
- The image dimensions ($W \times H$) are positive integers

File Format

The PPM P3 file format consists of:

- **Line 1:** Magic number "P3" indicating ASCII PPM format
- **Line 2:** Two integers separated by whitespace: width (W) and height (H)
- **Line 3:** Maximum color value (255 in this problem)
- **Lines 4 onwards:** RGB pixel values in row-major order. Each line contains $3 \times W$ integers representing the RGB values for one row of pixels, where each pixel consists of three integers (Red, Green, Blue) separated by whitespace

The pixel data contains $W \times H$ pixels, each with 3 color values (RGB), for a total of $3 \times W \times H$ integer values.

Sample Input

```
img.ppm
```

Sample Output

```
# "img_gray.pgm" saved sucessfully
```

Implementation

Goal: Fill in the following function:

```
from io import BufferedReader

def get_image(f: BufferedReader):
    ...
    return img


if __name__ == "__main__":
    import numpy as np
    input_filename = "img.ppm"
    with open(input_filename, "r") as f:
        img_matrix = np.array(get_image(f))
        assert img_matrix.max() <= 255 and img_matrix.min() >= 0, "Out of Bound"
    output_filename = "img_gray.pgm"
    gray_img = np.sum(img_matrix * np.array([0.299, 0.587, 0.114]), axis=-1)
    gray_img = gray_img.clip(0, 255).astype(np.uint8).astype(str)
    with open(output_filename, "w") as f:
        f.write("P2\n")
        f.write(f"{img_matrix.shape[0]} {img_matrix.shape[1]}\n")
        f.write("255\n")
        f.write(" ".join(np.ravel(gray_img)))
```

Reference

https://en.wikipedia.org/wiki/Netpbm

Warnings

- You are **NOT ALLOWED** to use the `numpy` library (except in the main block.)