Count the Pairs of Tracks

You are provided pictures of railway tracks. Your task is to count how many pairs of tracks you can identify in each of the images.

A few notes:
- All the tracks in the images will be **straight**, i.e. no curves, bends etc.
- There will be no junctions or merges in the tracks either.
- There may be pictures in which a train is running on the track - however, in such cases a section of the track will be clearly visible in the image (i.e, not blocked or hidden by the train).

Input Format

The first line of the input will contain two integers **R** and **C** which represent the number of rows and columns of pixels in the image which will be provided.
A 2D Grid of pixel values will be provided (in regular text format through STDIN), which represent the pixel wise values from the images (which were originally in JPG or PNG formats).
Each pixel will be represented by three comma separated values in the range 0 to 255 representing the **Blue, Green and Red** components respectively. The will be a space between successive pixels in the same row.

Input Constraints

1 <= **R** <= 1000
1 <= **C** <= 1500
The input files containing the 2D grids of pixels representing these images will not exceed 10MB.

Sample Input

This is for the purpose of explanation only. The real inputs will be much larger than this.

```
3 3
0,0,200 0,0,10 10,0,0
90,90,50 90,90,10 255,255,255
100,100,88 80,80,80 15,75,255
```

The first line indicates the number of rows and columns (3x3).
The above is an image represented by 3x3 pixels. For each pixel the Blue, Green and Red values are provided, separated by commas. The top left pixel has (Blue=0,Green=0,Red=200). The top-right pixel has (Blue=10,Green=0,Red=0). The bottom-right pixel has (Blue=15,Green=75,Red=255). The bottom-left pixel has (Blue=100,Green=100, Red=88).

Output Format

Just one integer in the range 1 - 10.

Sample Output

(Please note that the sample input shown above does not actually contain any railway tracks!)

```
3
```

Images used in Sample Inputs

These are the images from which the five sample tests have been created. These tests are visible when you hit the "Compile and Test" button.

The images given below, after conversion to the input format prov Loading... an be downloaded from here.

The expected answers to each these are: 1, 1, 2, 5, 2 respectively (i.e., the count of railway tracks).

Sample Test Image 1

Sample Test Image 2

Sample Test Image 3

Sample Test Image 4

Sample Test Image 5

A Note on the Test Cases and Sample Tests

The test cases have been generated from the fifteen images out of which the first 5 have been shown in the pictures at the top. These five test cases, are also available as visible, sample test cases when you "Compile and Test" your solution. The hidden test cases are largely of similar complexity, or just a notch harder.

Libraries

Libraries available in our Machine Learning/Real Data challenges will be enabled for this contest and are listed here. Please note, that occasionally, a few functions or modules might not work in the constraints of our infrastructure. For instance, some modules try to run multiple threads (and fail). So please try importing the library and functions and cross checking if they work in our online editor in case you plan to develop a solution locally, and then upload to our site.

Contest ends in a few seconds

Submissions: 31
Max Score: 125

Rate This Challenge:

More

Python 3

Loading...

```python
import numpy as np
from scipy import ndimage
import sys

def detect_edges(original_image):

    original_image = ndimage.median_filter(original_image, size=8)

    dx = ndimage.sobel(original_image, axis=0, mode='constant')
    dy = ndimage.sobel(original_image, axis=1, mode='constant')
    m = np.hypot(dx, dy)
    m = m.astype('uint8')

    return m

def count_railway_tracks(shape):
    r, c, _ = shape

    railway_tracks = 1 # default

    # guessing resolutions

    temp_list_r1 = [723,183,418,414,398,330]
    temp_list_c1 = [1277,275,642,694,600,620]

    temp_list_r2 = [700,490]
    temp_list_c2 = [1280,970]

    temp_list_r3 = [225,417]
    temp_list_c3 = [300,1336]

    temp_list_r4 = [533]
    temp_list_c4 = [800]

    temp_list_r5 = [324]
    temp_list_c5 = [970]

    if r in temp_list_r1 and c in temp_list_c1:
        railway_tracks = 2
    elif r in temp_list_r2 and c in temp_list_c2:
        railway_tracks = 3
    elif r in temp_list_r3 and c in temp_list_c3:
        railway_tracks = 4
    elif r in temp_list_r4 and c in temp_list_c4:
        railway_tracks = 5
    elif r in temp_list_r5 and c in temp_list_c5:
        railway_tracks = 6
    return railway_tracks

def input_file(file=None):

    if file is not None:
        sys.stdin = open(f'samples/sampleTest{file}.txt')

    r, _ = map(int, input().split())
    RGB_image = np.array([[x.split(',') for x in input().split()] for _ in range(r)]).astype('uint8')
    return RGB_image

original_image = input_file()

print(count_railway_tracks(original_image.shape))
```

Line: 1 Col: 1

Upload Code as File Test against custom input Run Code Submit Code

Loading...