

Logging Hell

Description

You are given a directory that contains arbitrarily deep subdirectory tree. Inside of the subdirectories scattered around you can find the log files (which end with ".logtxt" extension).

- Log files can be formatted in up to 5 different ways, you need to go through the files in the public dataset and figure out the exact formats.

Example of a directory tree containing .logtxt files.

- {main_directory}/{dirX}/{dirY}/file1.logtxt
- {main_directory}/{dirX}/{dirY}/file2.logtxt
- {main_directory}/{dirX}/{dirY}/file3.logtxt
- {main_directory}/{dirZ}/file2.logtxt
- {main_directory}/file0018.logtxt

These are the following tasks that you must do:

- A) Calculate the total number of ".logtxt" files
- B) Calculate the total number of log entries inside ".logtxt" files. (Each nonempty line inside a log is a log entry)
- C) Calculate the number of ".logtxt" files that have at least 1 error entry. (You need to deduce how exactly an error entry is described based on the given files)
- D) Calculate the 5 most common words that appear in the message body of each log entry at least once. (You need to deduce what exactly is the message body in different log formats based on the given files). Notes:
 - If multiple words occur with the same frequency, prioritize the one that appears first lexicographically.
 - If the total number of words is fewer than five, include all available words.
 - Consider all words case-sensitive.
- E) Find the longest period of time (in seconds) with at most 5 Warning log entries from the earliest log entry date to the latest log entry date. (You will need to know how the dates and Warning log entries are described in different log formats based on the given files). Note: Consider warning entries from all files!

Output format

Output answers to subtasks in 5 separate lines for each of the subtasks in order. If you don't have the answer to some answer leave a blank line.

- A) integer, total number of ".logtxt" files
- B) integer, total number of log entries
- C) integer, total number of ".logtxt" files that have at least 1 error entry
- D) string, comma separated list of 5 most common lexicographically sorted words (as explained in subtask D)
- E) integer, longest period of time (in seconds) with at most 5 warnings (as explained in subtask E)

Audio Video Sync

Description

The animations depict multiple circles, each varying in size, color, and speed, bouncing around the screen. With each bounce, a sound effect is generated. Between the bounces, the circles move linearly at a consistent speed. However, imagining these animations as taking place on a basketball court, you anticipate encountering pebbles and rough patches that might cause the circles to deviate in direction and speed upon impact.

Additionally, there is exactly one rectangular obstacle somewhere on the screen, off which the circles can bounce. You can assume that no circles will be inside this obstacle.

Circles can be different sizes, different colours and different speeds. You can assume they are not overlapping in the first frame, but they can freely move across one another during the video.

The audio file consists of a 1D array of numbers. The numbers represent the amount of noise being produced by the bounces continually, hidden in surrounding noise. Because the balls and impact points vary, the bounce sounds vary in intensity.

~~Audio signals are recorded with a sampling frequency of 44100Hz, and all video files have 60 FPS.~~ The technical specifications of your materials include a video frame rate of 60 FPS and an audio sampling frequency of 44100Hz. Given these resources, your tasks are as follows:

A) Return the center coordinates (row, column) of the left-most circle on the screen. The left-most circle is the one you would encounter first when coming from the left. You can assume that there will be only 1 left-most circle. *Note: notice that you should output the pixel coordinates in row-major order, i.e. the coordinate system starts in the top-left corner, with the x-axis running downwards.*

B) Return the center coordinates (row, column) of the rectangle obstacle.

C) Count the number of distinguishable bounces in the audio file (if two bounces perfectly overlap and cannot be distinguished count them as one bounce).

D) Count the number of bounces in the video stream. Assume that no shapes will be close to the edges near the start/end of the video.

E) Unfortunately, when you started your experiments, the video began playing before the audio, leading to mismatched start times. As a result, the audio and video files are not synchronized, with the video running ahead of the audio. Your task is to figure out where in the video the audio actually starts, and align them by printing the video frame number that corresponds to the start of the audio file.

Output format:

Output answers to subtasks in 5 separate lines for each of the subtasks in order. If you don't have the answer to some answer leave a blank line.

- A) two integers separated by a space representing left-most circle coordinates
- B) two integers separated by a space representing rectangle center coordinates
- C) a single integer representing number of distinguishable bounces in audio
- D) a single integer representing number of bounces in video
- E) a single integer representing the video frame number when the audio starts

Available packages

- `numpy`
- `imageio`
- all packages from the standard python library

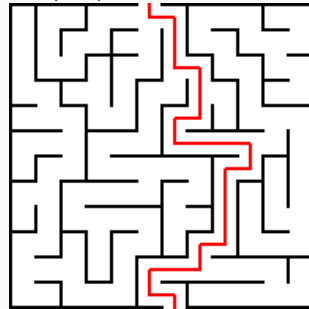
Additionally, feel free to use the simplified implementations of functions from `scipy.signal` available in the additional materials in the bottom of this task (`scipy_filter.zip`).

Maze

Description

Images can vary in size, but there are some general rules when it comes to how does the maze look like. Each pixel in the maze is either empty, hence free to step on or occupied by wall. Pixels that are not empty cannot be visited. Maze can also contain multiple gaps around the border that we call entrances. You can enter the maze in any one of them and exit in any other else. The goal is to find the shortest path one can traverse the maze on. Some definitions are given below.

There is one example of a maze and a valid path. Note that your path doesn't have to be *centered* like this one, you just can't stand on walls.



However, this maze is not just an ordinary maze. It can also contain a certain number of teleports. Those are special pixels that make you able to enter it and exit in any other teleport. You should also find the shortest path using teleports.

Task 1: Count entrances

Count the number of entrances in the given maze.

Task 2: Shortest path

Find the length of the shortest path in the maze.

Task 3: Shortest path with a twist

Find the length of the shortest path in the maze if you are allowed to use teleports.

Input

Input starts with a path to the image that contains maze. This image is guaranteed to be in PNG format. The next line contains the number of teleports N . The following N lines contain two numbers – row and column for each teleport in the maze.

Output

Output should contain three lines. The first line represents number of entrances in the maze. The second contains the length of the shortest path while the third contains the length of the shortest path if you are allowed to use teleports. Note that you are not required to use any teleport if the shortest path can be obtained without them. In case there is no valid path in the maze, output $-1-1$.

Definitions

- Pixel is said to be neighboring to another pixel if one can move either left, right, up, or down by exactly one step.
- Path is a list of neighboring pixels.
- The shortest path is a path that visits the least number of pixels.