2022 Fall CS101 Homework 2

Point Reflection

Total point: 50pts

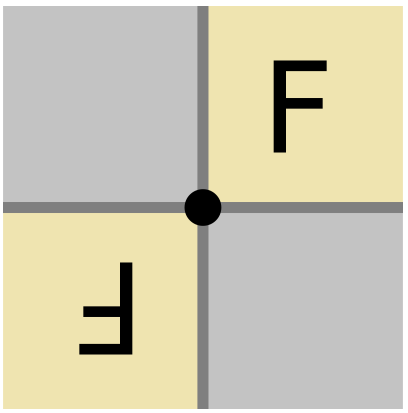
Out 04 Oct 2022
Due 11 Oct 2022

If you have any questions regarding Homework 2, use the ***Homework 2 Q&A*** board at Elice.

(Korean) 학습도우미 > 게시판 > Homework 2 Q&A
(English) Help Area > Forums > Homework 2 Q&A

Overview

Point reflection is a type of distance-preserving transformation of a figure where the figure is reflected through a single fixed point. In this homework, your task is to write functions for point reflection of images (Task 1 & 2) and the Hubo World (Task 3). You will start by implementing point reflection through the center point (Task 1) and then move onto point reflection through arbitrary points in the figure (Task 2 & 3).



Point reflection. (2022, September 6). In Wikipedia. https://en.wikipedia.org/wiki/Point_reflection

Task 1: Center Point Reflection of Images (10 points)

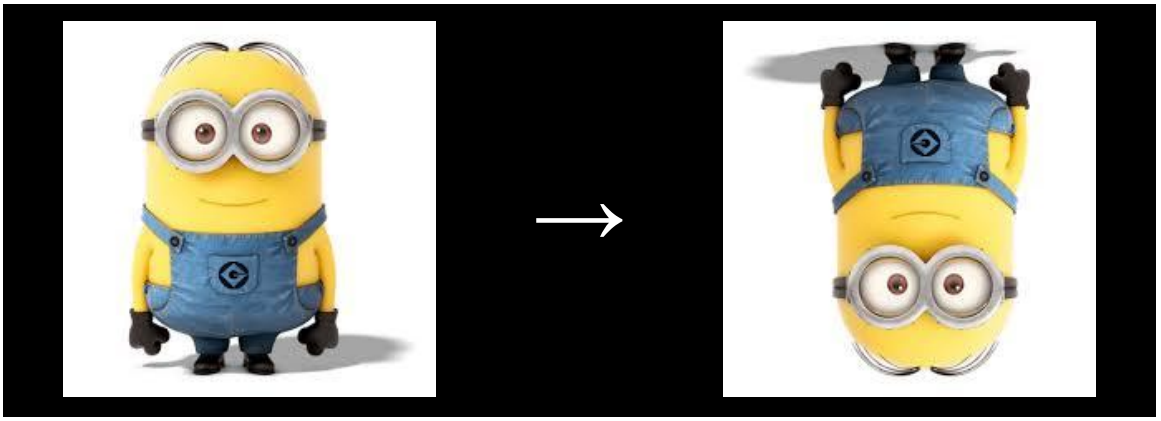
First, you will apply point reflection to images.
The figure below shows an example of point reflection through the center point of the image. In this case, the reflection is the same as a 180˚ rotation.

Implement the function named ***reflect_image*** under the following constraints.

- **Input**: An image of size $m \times n$, where m and n are positive integers.
- **Output**: An image with the same size ($m \times n$), which is the reflection of the input image with respect to the center point.

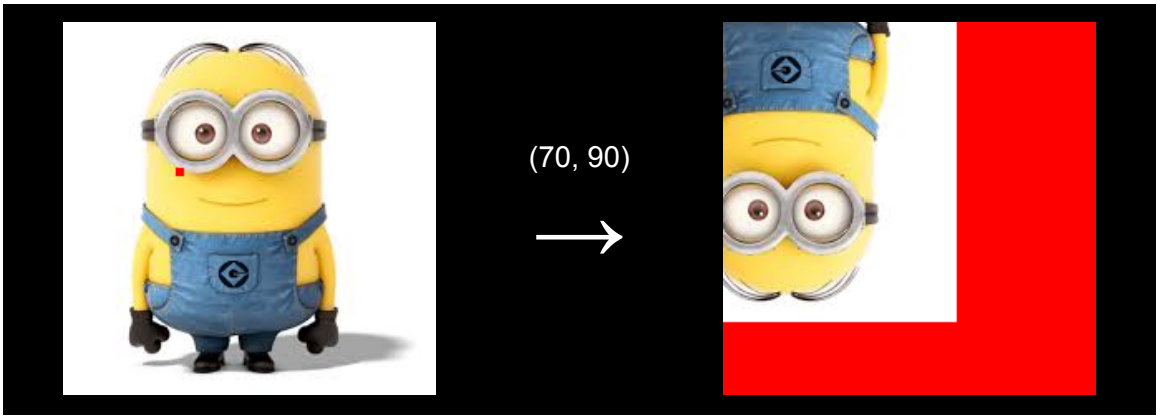
Tips

- Your function should work for any images of size $m \times n$, where each m and n can be either odd or even. You should be careful about locating the center of the image. When both m and n are odd, the center point is the $((m - 1) / 2, (n - 1) / 2)^{th}$ pixel. Otherwise, the center is located between pixels. For example, if m is even and n is odd, the center point is at the middle of the $(m / 2 - 1, (n - 1) / 2)^{th}$ pixel and the $(m / 2, (n - 1) / 2)^{th}$ pixel. (Notice that the indices start from 0.)



Task 2: Arbitrary Point Reflection of Images (20 points)

Now, you will extend your point reflection algorithm to be applied to any reference point other than the center point.
The figure below demonstrates an example of the arbitrary point reflection.
The left image is to be reflected with respect to the reference point indicated by the small red square around $(70, 90)^{th}$ pixel.
On the right is the resulting image cut to fit in the original space, with the remaining parts filled in red.



Implement the point reflection with respect to an arbitrary reference point on the input image, as illustrated in the following description.

Function Description

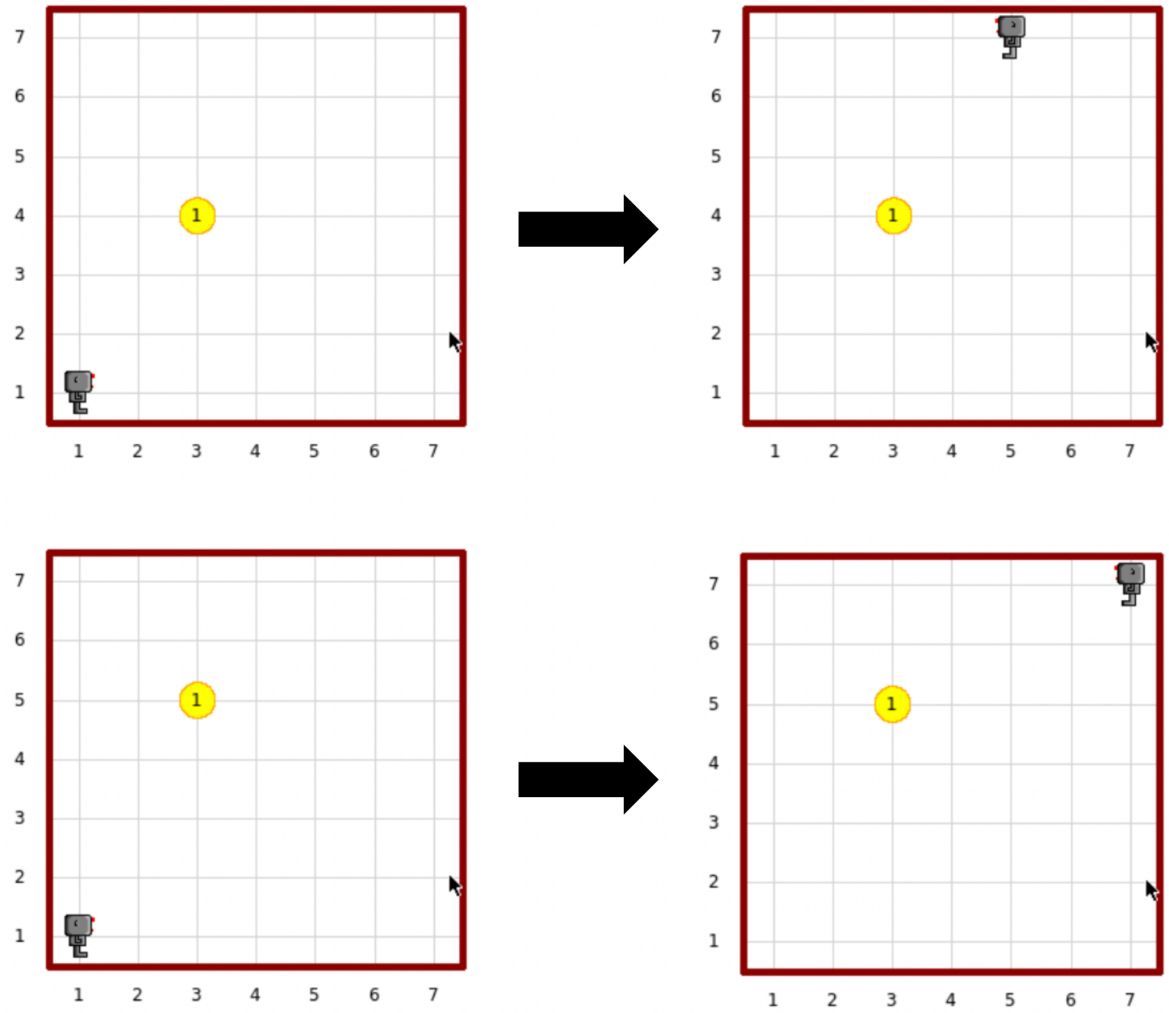
- Name
  - reflect_image
- Input
  - An image of size $m \times n$, where m and n are positive integers.
  - A reference point (a, b), where a and b are any integers.
- Output
  - The reflected image with respect to the reference point (a, b), of size m x n.
  - Cut off the image if the reflected pixel is out of the area of the original image.
  - Fill in the remaining area (i.e., the pixels which have no corresponding points in the original image) with red (RGB = (255, 0, 0))

Tips

- The $(0, 0)^{th}$ pixel is located in the upper left corner of the image.
- When a point outside the image is given as the reference point, you should return an image of the same size as the input image, completely filled with red.

Task 3: Point Reflection on Hubo World (20 points)

For the last task, write a function to move hubo to the position of the point reflection of the initial position centered on the position of a given beeper.
Hubo is initially on (1, 1) facing east. Find the position of the beeper which is located on an arbitrary point in the world. Then move hubo using point reflection. **If hubo's position after the point reflection is outside the world, leave hubo at the upper right corner.**



Implement the function **reflect_hubo** to move hubo based on point reflection.

- Hubo is initially on (1, 1) and facing east.
- The size of the given world is n x m where n and m are integers smaller than 10. There is only one beeper in the given world.
- Find the position of the beeper in the world and move hubo to the point reflected location of its initial position with respect to the beeper position.
- If hubo's position after the point reflection is outside the world, move hubo to the upper right corner (n, m) of the world.
- The direction of hubo after the reflection should also be reflected (facing west).