858. Mirror Reflection

Medium Geometry Math Number Theory

Leetcode Link

Problem Description

In this problem, we are presented with a unique situation involving a square room with mirrors on each of its four walls. Three of the room's corners have receptors numbered 0, 1, and 2. The fourth corner, located at the southwest, is where a laser ray originates. This ray first strikes the eastern wall at a certain distance `q` from the receptor numbered '0'. The wall length of the room is `p`.

We are tasked with determining which receptor the laser ray will hit first. The ray is guaranteed to eventually hit one of the receptors. To solve this problem, we can assume that the ray of light will continue to reflect off the mirrors indefinitely until it hits a receptor. The question boils down to analyzing the pattern of reflections to determine the point of contact with the right receptor.

The challenge lies in figuring out the conditions under which the ray will meet each receptor, which requires an understanding of the geometric properties of reflections within the square room.

Intuition

The solution to this problem involves recognizing a pattern in the reflections of the ray within the square room. A crucial observation is that the ray of light will periodically hit the same points on the walls after reflecting a certain number of times, effectively creating a "lattice" of potential intersection points within the room.

We start by calculating the greatest common divisor (gcd) of `p` and `q`, since the pattern of reflections will repeat every `p/gcd(p, q)` vertical distances and `q/gcd(p, q)` horizontal distances.

After finding the gcd, we divide both `p` and `q` by this value to reduce the problem to its simplest form - considering the minimal room size that still preserves the reflection pattern. Then we look at the parity (whether the number is even or odd) of `p/gcd(p, q)` and `q/gcd(p, q)`:

- If both `p/gcd(p, q)` and `q/gcd(p, q)` are odd, then the ray first meets the receptor at corner 1. After reflecting off the east wall, the ray needs an odd number of reflections to hit the north wall (which will point towards receptor 1), and an odd number of reflections off the north wall to return to the east wall at the height of receptor 1.
- If `p/gcd(p, q)` is odd but `q/gcd(p, q)` is even, the ray ends up at receptor 0. This is because the ray needs an odd number of reflections to go from east to north to west, and then an even number of reflections off the west wall will direct it back towards the south wall (opposite to receptor 0).
- Conversely, if `p/gcd(p, q)` is even but `q/gcd(p, q)` is odd, the ray first meets receptor 2. The ray will first travel up the east wall, reflect an even number of times horizontally, and end up on the east wall inline with receptor 2.

By determining the parity after dividing by the gcd, we can effectively predict which receptor will be hit first without simulating the entire path of the laser.

Solution Approach

The solution's approach hinges on the mathematical properties of reflection and the geometry of the square room in which the laser operates. The fundamental concept used in the implementation is that the path the laser will take before it hits a receptor will depend on the relationship between the distance the laser travels upwards (`q`) and the length of the wall (`p`). However, the complexity of this path is greatly simplified by examining the number of times these distances repeat themselves, which is deduced by their greatest common divisor (gcd).

Given `p` and `q`, we first determine their gcd using a function like `gcd(p, q)`. The `gcd` is useful because it allows us to reduce the problem to its simplest form where the path of the laser will hit one of the receptors without having to simulate its full path. By defining the smallest repeating pattern, it becomes easier to determine the first receptor it will reach.

Here's how the implementation unfolds step-by-step:

1. Calculate the gcd of `p` and `q`:
   ```
   g = gcd(p, q)
   ```
2. Divide `p` and `q` by the gcd to get the reduced number of vertical and horizontal reflections:
   ```
   p = (p // g) % 2
   q = (q // g) % 2
   ```
3. We then check the parity of `p` and `q` after this reduction:
   - If both `p` and `q` are odd (`p == 1 and q == 1`), the laser hits receptor 1.
   - If `p` is odd and `q` is even (`p == 1`), the laser hits receptor 0.
   - If `p` is even and `q` is odd (`q == 1`), the laser hits receptor 2.

Thus, the final part of the implementation is a simple map from these conditions to the receptor numbers:

```
if p == 1 and q == 1:
    return 1
return 0 if p == 1 else 2
```

This approach is successful because it abstracts away the infinite reflections and focuses on the essential behavior of the laser's path as defined by the number of intersections with the room's walls. The solution leverages the gcd to simplify the problem to a manageable algorithm without the need for intricate geometric calculations or simulations.

Example Walkthrough

Let's take a scenario where we have a square room and our task is to determine which receptor a laser beam will hit first. We are given that the room has a wall length `p = 8` and the laser strikes the eastern wall at a distance `q = 6` from receptor 0.

Following the steps of the solution approach, we will determine which receptor the laser hits:

1. Calculate the gcd of `p` and `q`:
   ```
   from math import gcd
   p = 8
   q = 6
   g = gcd(p, q)  # g = 2
   ```
2. Divide `p` and `q` by the gcd and determine their parity:
   ```
   p_reduced = (p // g) % 2  # (8 // 2) % 2 = 4 % 2 = 0 (even)
   q_reduced = (q // g) % 2  # (6 // 2) % 2 = 3 % 2 = 1 (odd)
   ```
3. Since `p_reduced` is even and `q_reduced` is odd, the laser hits receptor 2:
   ```
   if p_reduced == 1 and q_reduced == 1:
       receptor = 1
   else:
       receptor = 0 if p_reduced == 1 else 2

   print("The laser hits receptor", receptor)  # Outputs: The laser hits receptor: 2
   ```

The laser first strikes the eastern wall `6` units from receptor 0. After every two reflections (every `8` units it travels horizontally), the pattern repeats. Similarly, it strikes the same point on the horizontal axis every three vertical travels (every `6` units vertically). Therefore, after reducing `p` and `q` by `gcd(p, q)`, we find that the laser's path, given the reduced problem, would travel 4 units to the right and 3 units upward, both of which are effectively the same path repeated due to the gcd but with reduced multiples.

Since `p_reduced` is even, we know the laser will strike the east or west wall aligned with receptor 2. And because `q_reduced` is odd, it confirms that the laser is traveling upward from the lower wall, therefore, the laser hits the receptor numbered 2. The process of using the greatest common divisor simplifies the geometrical complexity and provides an elegant solution to the problem.

Python Solution

```
from math import gcd

class Solution:
    def mirrorReflection(self, p: int, q: int) -> int:
        # Calculate the greatest common divisor of p and q
        gcd_value = gcd(p, q)

        # Reduce p and q by their gcd and then find the parity (even or odd)
        # By taking each modulo 2.
        p_reduced = (p // gcd_value) % 2
        q_reduced = (q // gcd_value) % 2

        # A return value of 1 implies the light ray meets the requirement
        # to exit through the mirror labeled 1, which happens when both
        # p and q are odd after reduction and parity check.
        if p_reduced == 1 and q_reduced == 1:
            return 1

        # A return value of 0 means the light ray exits through the
        # mirror labeled 0, which occurs when p is odd (and thus q is even
        # since they cannot both be odd or both be even).
        # Otherwise, a return value of 2 is for when q is odd (mirror labeled 2).
        return 0 if p_reduced == 1 else 2

```

Java Solution

```
class Solution {
    // Method to find which receiver will get the laser signal
    public int mirrorReflection(int p, int q) {
        // Calculate the greatest common divisor of p and q
        int gcdValue = greatestCommonDivisor(p, q);

        // Even p means the laser hits the east wall (return 2)
        // Odd p means the laser hits the north/south wall
        // Odd q means the laser is at a corner position (return 1)
        // Even q means the laser is at the 0th receiver

        // Simplify the problem by dividing p and q by their gcd
        // and taking the parity (even or odd) of the result
        p = (p / gcdValue) % 2;
        q = (q / gcdValue) % 2;

        // If both p and q are odd, the laser will meet at receiver 1
        if (p == 1 && q == 1) {
            return 1;
        }
        // If p is odd and q is even, the laser will meet at receiver 0
        // If p is even and q is odd, the laser will meet at receiver 2
        return p == 1 ? 0 : 2;
    }

    // Helper method to calculate the greatest common divisor using the Euclidean algorithm
    private int greatestCommonDivisor(int a, int b) {
        // If b is 0, a is the gcd
        if (b == 0) {
            return a;
        }
        // Otherwise, recursively call the method with b and the remainder of a divided by b
        return greatestCommonDivisor(b, a % b);
    }
}
```

C++ Solution

```
#include <algorithm>  // std::gcd function is defined in the algorithm header

class Solution {
public:
    // The mirrorReflection function determines which of the three receptors (0, 1, or 2)
    // a laser beam will hit when it reflects off a square room's walls.
    int mirrorReflection(int p, int q) {
        // Compute the greatest common divisor (gcd) of p and q
        int gcd = std::gcd(p, q);

        // Divide p and q by their gcd to find their simplest ratio
        // then determining if each is odd or even by taking modulus 2
        // Evenness of p and q after division by gcd and taking remainder by 2
        p = (p / gcd) % 2;
        q = (q / gcd) % 2;

        // If p and q are both 1 (odd), the beam hits receptor 1
        if (p == 1 && q == 1) {
            return 1;
        }
        // If p is 1 (odd), the beam hits receptor 0
        // else (if q is 1, and p is 0 (even)), the beam hits receptor 2
        return p == 1 ? 0 : 2;
    }
};
```

Typescript Solution

```
// Function to calculate the position where the laser beam will meet the receptor
// after reflection in a room with dimensions p * p and a laser positioned at (p, 0),
// striking the wall at q * q units to the right of the corner.
// Returns 0, 1, or 2 corresponding to the three possible receptors' positions.
function mirrorReflection(p: number, q: number): number {
    // Find the greatest common divisor of p and q
    const gcdValue: number = gcd(p, q);
    // Evenness of p and q after division by gcd and taking remainder by 2
    p %= 2;
    q %= 2;

    // When both are odd, the beam meets at receptor 1
    if (p === 1 && q === 1) {
        return 1;
    }
    // When p is odd, the beam meets at receptor 0, else at receptor 2
    return p === 1 ? 0 : 2;
}

// Helper function to find the greatest common divisor (GCD)
// of two numbers using the Euclidean algorithm
function gcd(a: number, b: number): number {
    // If b is 0, a is the GCD.
    if (b === 0) {
        return a;
    }
    // Recursively call the gcd function with b and a % b until b equals 0
    return gcd(b, a % b);
}
```

Time and Space Complexity

The time complexity of the code is `O(log(min(p, q)))` where `p` is the length of one side of the square and `q` is the length of the extension where the laser hits. This time complexity arises from the call to the Greatest Common Divisor (`gcd`) function, which is typically implemented using the Euclidean algorithm. The Euclidean algorithm's time complexity is a logarithmic function of the input values, hence `O(log(min(p, q)))`.

The space complexity of the code is `O(1)` because the space required does not scale with the size of the input. Only a fixed number of integer variables (`g`, `p`, `q`) are used, which occupy constant space.