Problem 3609: Minimum Moves to Reach Target in Grid

Problem Information

Difficulty: Hard
Acceptance Rate: 14.34%
Paid Only: No
Tags: Math

Problem Description

You are given four integers `sx`, `sy`, `tx`, and `ty`, representing two points `(sx, sy)` and `(tx, ty)` on an infinitely large 2D grid.

You start at `(sx, sy)`.

At any point `(x, y)`, define `m = max(x, y)`. You can either:

* Move to `(x + m, y)`, or * Move to `(x, y + m)`.

Return the **minimum** number of moves required to reach `(tx, ty)`. If it is impossible to reach the target, return -1.

Example 1:

Input: sx = 1, sy = 2, tx = 5, ty = 4

Output: 2

Explanation:

The optimal path is:

* Move 1: `max(1, 2) = 2`. Increase the y-coordinate by 2, moving from `(1, 2)` to `(1, 2 + 2) = (1, 4)`. * Move 2: `max(1, 4) = 4`. Increase the x-coordinate by 4, moving from `(1, 4)` to `(1 +

4, 4) = (5, 4)`.

Thus, the minimum number of moves to reach `(5, 4)` is 2.

Example 2:

Input: sx = 0, sy = 1, tx = 2, ty = 3

Output: 3

Explanation:

The optimal path is:

* Move 1: `max(0, 1) = 1`. Increase the x-coordinate by 1, moving from `(0, 1)` to `(0 + 1, 1) = (1, 1)`. * Move 2: `max(1, 1) = 1`. Increase the x-coordinate by 1, moving from `(1, 1)` to `(1 + 1, 1) = (2, 1)`. * Move 3: `max(2, 1) = 2`. Increase the y-coordinate by 2, moving from `(2, 1)` to `(2, 1 + 2) = (2, 3)`.

Thus, the minimum number of moves to reach `(2, 3)` is 3.

Example 3:

Input: sx = 1, sy = 1, tx = 2, ty = 2

Output: -1

Explanation:

* It is impossible to reach `(2, 2)` from `(1, 1)` using the allowed moves. Thus, the answer is -1.

Constraints:

* `0 <= sx <= tx <= 109` * `0 <= sy <= ty <= 109`

Code Snippets

C++:

```
class Solution {
public:
int minMoves(int sx, int sy, int tx, int ty) {

}
};
```

Java:

```
class Solution {
public int minMoves(int sx, int sy, int tx, int ty) {

}
}
```

Python3:

```
class Solution:
def minMoves(self, sx: int, sy: int, tx: int, ty: int) -> int:
```