874. Walking Robot Simulation

Solved ●

Medium Topics Companies

A robot on an infinite XY-plane starts at point `(0, 0)` facing north. The robot receives an array of integers `commands`, which represents a sequence of moves that it needs to execute. There are only three possible types of instructions the robot can receive:

- `-2`: Turn left `90` degrees.
- `-1`: Turn right `90` degrees.
- `1 <= k <= 9`: Move forward `k` units, one unit at a time.

Some of the grid squares are `obstacles`. The i^{th} obstacle is at grid point `obstacles[i] = (x_i, y_i)`. If the robot runs into an obstacle, it will stay in its current location (on the block adjacent to the obstacle) and move onto the next command.

Return the **maximum squared Euclidean distance** that the robot reaches at any point in its path (i.e. if the distance is `5`, return `25`).

Note:

- There can be an obstacle at `(0, 0)`. If this happens, the robot will ignore the obstacle until it has moved off the origin. However, it will be unable to return to `(0, 0)` due to the obstacle.
- North means +Y direction.
- East means +X direction.
- South means -Y direction.
- West means -X direction.

Example 1:

> **Input:** commands = [4,-1,3], obstacles = []
>
> **Output:** 25
>
> **Explanation:**
>
> The robot starts at `(0, 0)`:
>
> 1. Move north 4 units to `(0, 4)`.
> 2. Turn right.
> 3. Move east 3 units to `(3, 4)`.
>
> The furthest point the robot ever gets from the origin is `(3, 4)`, which squared is $3^2 + 4^2 = 25$ units away.

Example 2:

> **Input:** commands = [4,-1,4,-2,4], obstacles = [[2,4]]
>
> **Output:** 65
>
> **Explanation:**
>
> The robot starts at `(0, 0)`:
>
> 1. Move north 4 units to `(0, 4)`.
> 2. Turn right.
> 3. Move east 1 unit and get blocked by the obstacle at `(2, 4)`, robot is at `(1, 4)`.

> 4. Turn left.
> 5. Move north 4 units to $(1, 8)$.
>
> The furthest point the robot ever gets from the origin is $(1, 8)$, which squared is $1^2 + 8^2 = 65$ units away.

Example 3:

> **Input:** commands = [6,-1,-1,6], obstacles = [[0,0]]
>
> **Output:** 36
>
> **Explanation:**
>
> The robot starts at $(0, 0)$:
>
> 1. Move north 6 units to $(0, 6)$.
> 2. Turn right.
> 3. Turn right.
> 4. Move south 5 units and get blocked by the obstacle at $(0,0)$, robot is at $(0, 1)$.
>
> The furthest point the robot ever gets from the origin is $(0, 6)$, which squared is $6^2 = 36$ units away.

Constraints:

- $1 <= commands.length <= 10^4$
- `commands[i]` is either -2, -1, or an integer in the range $[1, 9]$.
- $0 <= obstacles.length <= 10^4$
- $-3 * 10^4 <= x_i, y_i <= 3 * 10^4$
- The answer is guaranteed to be less than 2^{31}.

Seen this question in a real interview before? 1/5

Yes No

Accepted **149.7K** Submissions **259.4K** Acceptance Rate **57.7%**

Topics

Companies

Similar Questions

Discussion (167)