Problem 335: Self Crossing

Problem Information

Difficulty: Hard
Acceptance Rate: 33.59%
Paid Only: No
Tags: Array, Math, Geometry

Problem Description

You are given an array of integers `distance`.

You start at the point `(0, 0)` on an **X-Y plane,** and you move `distance[0]` meters to the north, then `distance[1]` meters to the west, `distance[2]` meters to the south, `distance[3]` meters to the east, and so on. In other words, after each move, your direction changes counter-clockwise.

Return `true` _if your path crosses itself or_`false` _if it does not_.

Example 1:

![](https://assets.leetcode.com/uploads/2022/12/21/11.jpg)

Input: distance = [2,1,1,2] **Output:** true **Explanation:** The path crosses itself at the point (0, 1).

Example 2:

![](https://assets.leetcode.com/uploads/2022/12/21/22.jpg)

Input: distance = [1,2,3,4] **Output:** false **Explanation:** The path does not cross itself at any point.

Example 3:

![](https://assets.leetcode.com/uploads/2022/12/21/33.jpg)

Input: distance = [1,1,1,2,1] **Output:** true **Explanation:** The path crosses itself at the point (0, 0).

Constraints:

* `1 <= distance.length <= 105` * `1 <= distance[i] <= 105`

Code Snippets

C++:

```
class Solution {
public:
bool isSelfCrossing(vector<int>& distance) {

}
};
```

Java:

```
class Solution {
public boolean isSelfCrossing(int[] distance) {

}
}
```

Python3:

```
class Solution:
def isSelfCrossing(self, distance: List[int]) -> bool:
```