Problem 1496: Path Crossing

Problem Information

Difficulty: Easy
Acceptance Rate: 62.51%
Paid Only: No
Tags: Hash Table, String

Problem Description

Given a string `path`, where `path[i] = 'N'`, `'S'`, `'E'` or `'W'`, each representing moving one unit north, south, east, or west, respectively. You start at the origin `(0, 0)` on a 2D plane and walk on the path specified by `path`.

Return `true` _if the path crosses itself at any point, that is, if at any time you are on a location you have previously visited_. Return `false` otherwise.

Example 1:

![](https://assets.leetcode.com/uploads/2020/06/10/screen-shot-2020-06-10-at-123929-pm.png)

Input: path = "NES" **Output:** false **Explanation:** Notice that the path doesn't cross any point more than once.

Example 2:

![](https://assets.leetcode.com/uploads/2020/06/10/screen-shot-2020-06-10-at-123843-pm.png)

Input: path = "NESWW" **Output:** true **Explanation:** Notice that the path visits the origin twice.

Constraints:

* `1 <= path.length <= 104` * `path[i]` is either `'N'`, `'S'`, `'E'`, or `'W'`.

Code Snippets

C++:

```
class Solution {
public:
bool isPathCrossing(string path) {

}
};
```

Java:

```
class Solution {
public boolean isPathCrossing(String path) {

}
}
```

Python3:

```
class Solution:
def isPathCrossing(self, path: str) -> bool:
```