Problem 3694: Distinct Points Reachable After Substring Removal

Problem Information

Difficulty: Medium
Acceptance Rate: 53.13%
Paid Only: No
Tags: Hash Table, String, Sliding Window, Prefix Sum

Problem Description

You are given a string `s` consisting of characters `'U'`, `'D'`, `'L'`, and `'R'`, representing moves on an infinite 2D Cartesian grid.

* `'U'`: Move from `(x, y)` to `(x, y + 1)`. * `'D'`: Move from `(x, y)` to `(x, y - 1)`. * `'L'`: Move from `(x, y)` to `(x - 1, y)`. * `'R'`: Move from `(x, y)` to `(x + 1, y)`.

You are also given a positive integer `k`.

You **must** choose and remove **exactly one** contiguous substring of length `k` from `s`. Then, start from coordinate `(0, 0)` and perform the remaining moves in order.

Return an integer denoting the number of **distinct** final coordinates reachable.

Example 1:

Input: s = "LUL", k = 1

Output: 2

Explanation:

After removing a substring of length 1, `s` can be `"UL"`, `"LL"` or `"LU"`. Following these moves, the final coordinates will be `(-1, 1)`, `(-2, 0)` and `(-1, 1)` respectively. There are two distinct points `(-1, 1)` and `(-2, 0)` so the answer is 2.

Example 2:

Input: s = "UDLR", k = 4

Output: 1

Explanation:

After removing a substring of length 4, `s` can only be the empty string. The final coordinates will be `(0, 0)`. There is only one distinct point `(0, 0)` so the answer is 1.

Example 3:

Input: s = "UU", k = 1

Output: 1

Explanation:

After removing a substring of length 1, `s` becomes `"U"`, which always ends at `(0, 1)`, so there is only one distinct final coordinate.

Constraints:

* `1 <= s.length <= 105` * `s` consists of only `'U'`, `'D'`, `'L'`, and `'R'`. * `1 <= k <= s.length`

Code Snippets

C++:

```
class Solution {
public:
int distinctPoints(string s, int k) {

}
};
```

Java:

```
class Solution {
public int distinctPoints(String s, int k) {

}
}
```

Python3:

```
class Solution:
def distinctPoints(self, s: str, k: int) -> int:
```