Problem 942: DI String Match

Problem Information

Difficulty: Easy
Acceptance Rate: 80.64%
Paid Only: No
Tags: Array, Two Pointers, String, Greedy

Problem Description

A permutation `perm` of `n + 1` integers of all the integers in the range `[0, n]` can be represented as a string `s` of length `n` where:

* `s[i] == 'I'` if `perm[i] < perm[i + 1]`, and * `s[i] == 'D'` if `perm[i] > perm[i + 1]`.

Given a string `s`, reconstruct the permutation `perm` and return it. If there are multiple valid permutations perm, return **any of them**.

Example 1:

Input: s = "IDID" **Output:** [0,4,1,3,2]

Example 2:

Input: s = "III" **Output:** [0,1,2,3]

Example 3:

Input: s = "DDI" **Output:** [3,2,0,1]

Constraints:

* `1 <= s.length <= 105` * `s[i]` is either `'I'` or `'D'`.

Code Snippets

C++:

```
class Solution {
public:
vector<int> diStringMatch(string s) {

}
};
```

Java:

```
class Solution {
public int[] diStringMatch(String s) {

}
}
```

Python3:

```
class Solution:
def diStringMatch(self, s: str) -> List[int]:
```