

Problem 484: Find Permutation

Problem Information

Difficulty: Medium

Acceptance Rate: 66.93%

Paid Only: Yes

Tags: Array, String, Stack, Greedy

Problem Description

A permutation `perm` of `n` integers of all the integers in the range `[1, n]` can be represented as a string `s` of length `n - 1` 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 lexicographically smallest permutation `perm` and return it.

Example 1:

Input: `s = "I"` **Output:** `[1,2]` **Explanation:** `[1,2]` is the only legal permutation that can be represented by `s`, where the number 1 and 2 construct an increasing relationship.

Example 2:

Input: `s = "DI"` **Output:** `[2,1,3]` **Explanation:** Both `[2,1,3]` and `[3,1,2]` can be represented as `"DI"`, but since we want to find the smallest lexicographical permutation, you should return `[2,1,3]`

Constraints:

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

Code Snippets

C++:

```
class Solution {  
public:  
    vector<int> findPermutation(string s) {  
  
    }  
};
```

Java:

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

Python3:

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