Problem 903: Valid Permutations for DI Sequence

Problem Information

Difficulty: Hard
Acceptance Rate: 56.01%
Paid Only: No
Tags: String, Dynamic Programming, Prefix Sum

Problem Description

You are given a string `s` of length `n` where `s[i]` is either:

* `'D'` means decreasing, or * `'I'` means increasing.

A permutation `perm` of `n + 1` integers of all the integers in the range `[0, n]` is called a **valid permutation** if for all valid `i`:

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

Return _the number of**valid permutations** _`perm`. Since the answer may be large, return it **modulo** `109 + 7`.

Example 1:

Input: s = "DID" **Output:** 5 **Explanation:** The 5 valid permutations of (0, 1, 2, 3) are: (1, 0, 3, 2) (2, 0, 3, 1) (2, 1, 3, 0) (3, 0, 2, 1) (3, 1, 2, 0)

Example 2:

Input: s = "D" **Output:** 1

Constraints:

* `n == s.length` * `1 <= n <= 200` * `s[i]` is either `'I'` or `'D'`.

Code Snippets

C++:

```
class Solution {
public:
int numPermsDISequence(string s) {

}
};
```

Java:

```
class Solution {
public int numPermsDISequence(String s) {

}
}
```

Python3:

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