Problem 2375: Construct Smallest Number From DI String

Problem Information

Difficulty: Medium
Acceptance Rate: 85.67%
Paid Only: No
Tags: String, Backtracking, Stack, Greedy

Problem Description

You are given a **0-indexed** string `pattern` of length `n` consisting of the characters `'I'` meaning **increasing** and `'D'` meaning **decreasing**.

A **0-indexed** string `num` of length `n + 1` is created using the following conditions:

* `num` consists of the digits `'1'` to `'9'`, where each digit is used **at most** once. * If `pattern[i] == 'I'`, then `num[i] < num[i + 1]`. * If `pattern[i] == 'D'`, then `num[i] > num[i + 1]`.

Return _the lexicographically**smallest** possible string _`num` _that meets the conditions._

Example 1:

Input: pattern = "IIIDIDDD" **Output:** "123549876" **Explanation:** At indices 0, 1, 2, and 4 we must have that num[i] < num[i+1]. At indices 3, 5, 6, and 7 we must have that num[i] > num[i+1]. Some possible values of num are "245639871", "135749862", and "123849765". It can be proven that "123549876" is the smallest possible num that meets the conditions. Note that "123414321" is not possible because the digit '1' is used more than once.

Example 2:

Input: pattern = "DDD" **Output:** "4321" **Explanation:** Some possible values of num are "9876", "7321", and "8742". It can be proven that "4321" is the smallest possible num that meets the conditions.

Constraints:

* `1 <= pattern.length <= 8` * `pattern` consists of only the letters `'I'` and `'D'`.

Code Snippets

C++:

```
class Solution {
public:
string smallestNumber(string pattern) {

}
};
```

Java:

```
class Solution {
public String smallestNumber(String pattern) {

}
}
```

Python3:

```
class Solution:
def smallestNumber(self, pattern: str) -> str:
```