

2375. Construct Smallest Number From DI String

Medium Topics Companies Hint

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 = "IIDIIDDD"
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.

1 2 3 5 4 9 8 7 6
→ I I I D I D D D
I: 4 D: 4

if pattern[0] == "I"
res[0] = 1
else

res[0] = # of first 'D's + 1

len = 5
D D D I D
6 5 4 3 2 1 x
4 3 2 1 6 5 ✓
m = len(pattern)
avail-hu = set{1, 2, ..., m, m+1}
1°
2°

Greedy

... D I D ...
seen = set(...)

I : n+1th smallest available

... I D I ...
n

n+1th smallest available

Description
Accepted
Editorial
Solutions
Submissions

All Submissions

Accepted 104 / 104 testcases passed
AndrewC275 submitted at Feb 18, 2025 10:26

Editorial
Solution

Runtime
0 ms | Beats 100.00%

Memory
17.76 MB | Beats 54.04%

Code
Python3
Auto

```

1 class Solution:
2     def smallestNumber(self, pattern: str) -> str:
3         res, stack = [], []
4         for i in range(len(pattern)+1):
5             stack.append(i+1)
6
7         while stack and (i == len(pattern) or pattern[i] == "I"):
8             res.append(str(stack.pop()))
9
10        return "".join(res)
11
12

```

Ln 8, Col 45 | Saved

Testcase
Test Result

Accepted Runtime: 0 ms

Case 1
Case 2

eg. DIIIIDD len = 7

	stack = [1]	res = []	stack [1]
i =>			
1	[1, 2]	[2, 1]	[]
2	[3]	[2, 1, 3]	[]
3	[4]	[2, 1, 3, 4]	[]
4	[5]	[2, 1, 3, 4, 5]	[]
5	[6]	[2, 1, 3, 4, 5]	[6]
6	[6, 7]	[2, 1, 3, 4, 5, 7, 6]	[]