Problem 1849: Splitting a String Into Descending Consecutive Values

Problem Information

Difficulty: Medium
Acceptance Rate: 37.21%
Paid Only: No
Tags: String, Backtracking, Enumeration

Problem Description

You are given a string `s` that consists of only digits.

Check if we can split `s` into **two or more non-empty substrings** such that the **numerical values** of the substrings are in **descending order** and the **difference** between numerical values of every two **adjacent** **substrings** is equal to `1`.

* For example, the string `s = "0090089"` can be split into `["0090", "089"]` with numerical values `[90,89]`. The values are in descending order and adjacent values differ by `1`, so this way is valid. * Another example, the string `s = "001"` can be split into `["0", "01"]`, `["00", "1"]`, or `["0", "0", "1"]`. However all the ways are invalid because they have numerical values `[0,1]`, `[0,1]`, and `[0,0,1]` respectively, all of which are not in descending order.

Return `true` _if it is possible to split_ `s`■■■■■■ _as described above_ _, or_`false` _otherwise._

A **substring** is a contiguous sequence of characters in a string.

Example 1:

Input: s = "1234" **Output:** false **Explanation:** There is no valid way to split s.

Example 2:

Input: s = "050043" **Output:** true **Explanation:** s can be split into ["05", "004", "3"] with numerical values [5,4,3]. The values are in descending order with adjacent values

differing by 1.

Example 3:

Input: s = "9080701" **Output:** false **Explanation:** There is no valid way to split s.

Constraints:

* `1 <= s.length <= 20` * `s` only consists of digits.

Code Snippets

C++:

```
class Solution {
public:
bool splitString(string s) {

}
};
```

Java:

```
class Solution {
public boolean splitString(String s) {

}
}
```

Python3:

```
class Solution:
def splitString(self, s: str) -> bool:
```