Count and Say Sequence Documentation

Overview

The Count and Say sequence is a sequence of digit strings defined by a recursive formula. It is characterized by converting the previous term into a different digit string based on how the digits are spoken or counted.

Formula

The recursive formula for the Count and Say sequence is as follows:

- `countAndSay(1) = "1"`
- `countAndSay(n)` is the way you would "say" the digit string from `countAndSay(n-1)`, which is then converted into a different digit string.

Methodology

To determine how a digit string is "said," follow these steps:

1. Split the digit string into the minimal number of substrings such that each substring contains exactly one unique digit.
2. For each substring, count the number of digits and say the digit.
3. Concatenate every said digit.

Example

For example, let's take the digit string "3322251":

- Splitting it, we get substrings: "33", "222", "5", "1".
- Counting and saying each substring, we get: "two 3s", "three 2s", "one 5", "one 1".
- Concatenating these, we get the digit string "23321511".

Implementation

The provided solution is implemented in Python using a recursive approach:

```
class Solution:
    def countAndSay(self, n: int) -> str:
        if n == 1:
            return "1"
        prev = self.countAndSay(n - 1)
        result = ""
        count = 1
        for i in range(len(prev)):
            if i + 1 < len(prev) and prev[i] == prev[i + 1]:
                count += 1
            else:
                result += str(count) + prev[i]
                count = 1
        return result
```

This solution recursively calculates the Count and Say sequence up to the nth term based on the provided input integer `n`. It iterates through the previous term's string, counts consecutive occurrences of each digit, and constructs the new digit string accordingly. Finally, it returns the resulting digit string for the nth term.

Example Usage

Example 1:

n = 1

print(Solution().countAndSay(n)) # Output: "1"

Example 2:

n = 4

print(Solution().countAndSay(n)) # Output: "1211"

Constraints

- The input integer `n` lies within the range 1 to 30, inclusive.