

## .Count and Say

The count-and-say sequence is a sequence of digit strings defined by the recursive formula:

- `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.

To determine how you "say" a digit string, split it into the minimal number of substrings such that each substring contains exactly one unique digit. Then for each substring, say the number of digits, then say the digit. Finally, concatenate every said digit.

For example, the saying and conversion for digit string "3322251":

Given a positive integer *n*, return the *n*th term of the count-and-say sequence.

Example 1:

Input: *n* = 1

Output: "1"

Explanation: This is the base case.

Example 2:

Input: *n* = 4

Output: "1211"

Explanation:

`countAndSay(1) = "1"`

`countAndSay(2) = say "1" = one 1 = "11"`

`countAndSay(3) = say "11" = two 1's = "21"`

`countAndSay(4) = say "21" = one 2 + one 1 = "12" + "11" = "1211"`

Program:

```
ddef countAndSay(n):
    if n == 1:
        return "1"
    def next_sequence(seq):
        result = []
        i = 0
        while i < len(seq):
            count = 1
            while i + 1 < len(seq) and seq[i] == seq[i + 1]:
                i += 1
                count += 1
            result.append(str(count) + seq[i])
            i += 1
        return "".join(result)
    current = "1"
    for _ in range(1, n):
        current = next_sequence(current)
    return current
n = 1
print(countAndSay(n))
```

```
n = 4
print(countAndSay(n))
```

Output:

```
C:\Users\srika\Desktop\CSA0863\pythonProject\.venv\Scripts\python.exe "C:\Users\srika\Desktop\CSA0863\pythonProject\DAACOAADS.PYTHON\PROGRAM_69.PY"
1
1211

Process finished with exit code 0
```

Time complexity:  
 $O(2^n)$