

38. Count and Say

Solved

Medium Topics Companies Hint

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

- `countAndSay(1) = "1"`
- `countAndSay(n)` is the run-length encoding of `countAndSay(n - 1)`.

Run-length encoding (RLE) is a string compression method that works by replacing consecutive identical characters (repeated 2 or more times) with the concatenation of the character and the number marking the count of the characters (length of the run). For example, to compress the string "3322251" we replace "33" with "23", replace "222" with "32", replace "5" with "15" and replace "1" with "11". Thus the compressed string becomes "23321511".

Given a positive integer `n`, return the n^{th} element of the **count-and-say** sequence.

Example 1:

Input: `n = 4`

Output: "1211"

Explanation:

`countAndSay(1) = "1"`

`countAndSay(2) = RLE of "1" = "11"`

Problem List < > 🔍 Run Submit 🗑️ 📄 Premium

Description Accepted Editorial Solutions Submissions

38. Count and Say

Solved

Medium Topics Companies Hint

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

- `countAndSay(1) = "1"`
- `countAndSay(n)` is the run-length encoding of `countAndSay(n - 1)`.

Run-length encoding (RLE) is a string compression method that works by replacing consecutive identical characters (repeated 2 or more times) with the concatenation of the character and the number marking the count of the characters (length of the run). For example, to compress the string "3322251" we replace "33" with "23", replace "222" with "32", replace "5" with "15" and replace "1" with "11". Thus the compressed string becomes "23321511".

Given a positive integer `n`, return the n^{th} element of the **count-and-say** sequence.

Example 1:

Input: `n = 4`

Output: "1211"

Explanation:

`countAndSay(1) = "1"`
`countAndSay(2) = RLE of "1" = "11"`
`countAndSay(3) = RLE of "11" = "21"`
`countAndSay(4) = RLE of "21" = "1211"`

4K 129

Code

```
C#
1 using System;
2 using System.Text;
3
4 public class Solution {
5     public string CountAndSay(int n) {
6         if (n <= 0) {
7             return "";
8         }
9         string result = "1";
10
11     }
```

Saved Ln 43, Col 1

Testcase **Test Result**

Accepted Runtime: 50 ms

Case 1 Case 2

Input

`n =`

1

Output

"1"

Expected

Код:

```
using System;
using System.Text;

public class Solution
{
    public string CountAndSay(int n)
```

```

{
    if (n <= 0)
    {
        return "";
    }

    string result = "1";

    for (int i = 2; i <= n; i++)
    {
        result = GetNextSequence(result);
    }

    return result;
}

private string GetNextSequence(string sequence)
{
    StringBuilder nextSequence = new StringBuilder();

    int count = 1;
    char currentChar = sequence[0];

    for (int i = 1; i < sequence.Length; i++)
    {
        if (sequence[i] == currentChar)
        {
            count++;
        }
        else
        {
            nextSequence.Append(count);
            nextSequence.Append(currentChar);
            currentChar = sequence[i];
            count = 1;
        }
    }

    // Append the last run
    nextSequence.Append(count);
    nextSequence.Append(currentChar);

    return nextSequence.ToString();
}
}

```