# Dry Run:

|  |  |  |  |
| --- | --- | --- | --- |
| **n** | **Input Term** | **Grouping (Count & Char)** | **Output Term (buildNext)** |
| 1 | "1" | one 1 | "11" |
| 2 | "11" | two 1s | "21" |
| 3 | "21" | one 2, one 1 | "1211" |
| 4 | "1211" | one 1, one 2, two 1s | "111221" |
| 5 | "111221" | three 1s, two 2s, one 1 | "312211" |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **i** | **currentChar** | **prevChar** | **count** | **Action** | **sb (StringBuilder so far)** |
| 1 | '2' | '1' | 1 | Append "1" + '1' | "11" |
|  |  | Update → prevChar = '2', count = 1 |  |  |  |
| 2 | '1' | '2' | 1 | Append "1" + '2' | "1112" |
|  |  | Update → prevChar = '1', count = 1 |  |  |  |
| 3 | '1' | '1' | 2 | Just increment count | "1112" |
| — | end of loop |  |  | Append "2" + '1' (last group) | "111221" |

# Solution:

public class Solution {

    public String countAndSay(int n) {

        if (n <= 0) return "";

        String result = "1"; // Starting point: countAndSay(1)

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

            result = buildNext(result);

        }

        return result;

    }

    private String buildNext(String s) {

        StringBuilder sb = new StringBuilder();

        int count = 1;

        char prevChar = s.charAt(0);

        for (int i = 1; i < s.length(); i++) {

            char currentChar = s.charAt(i);

            if (currentChar == prevChar) {

                count++;

            } else {

                sb.append(count).append(prevChar); // say the count and the digit

                count = 1;

                prevChar = currentChar;

            }

        }

        // Append the last group

        sb.append(count).append(prevChar);

        return sb.toString();

    }

}