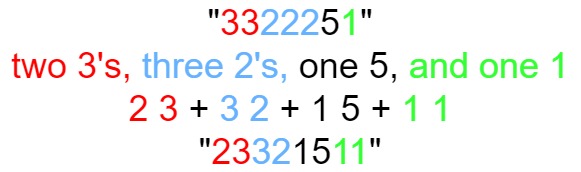
The **count-and-say** sequence is a sequence of digit strings defined by the recursive formula:

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 groups so that each group is a contiguous section all of the **same character.** Then for each group, say the number of characters, then say the character. To convert the saying into a digit string, replace the counts with a number and concatenate every saying.

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



Given a positive integer n, return *the*nth*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"

Solution:

class Solution {

public String countAndSay(int n) {

if(n==1) return "1";

if(n==2) return "11";

String s = "11";

for(int i=3;i<=n;i++){

String t="";

s+='&';

int c=1;

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

if(s.charAt(j) != s.charAt(j-1)){

t=t+c;

t+= s.charAt(j-1);

c=1;

}

else c++;

}

s=t;

}

return s;

}

}