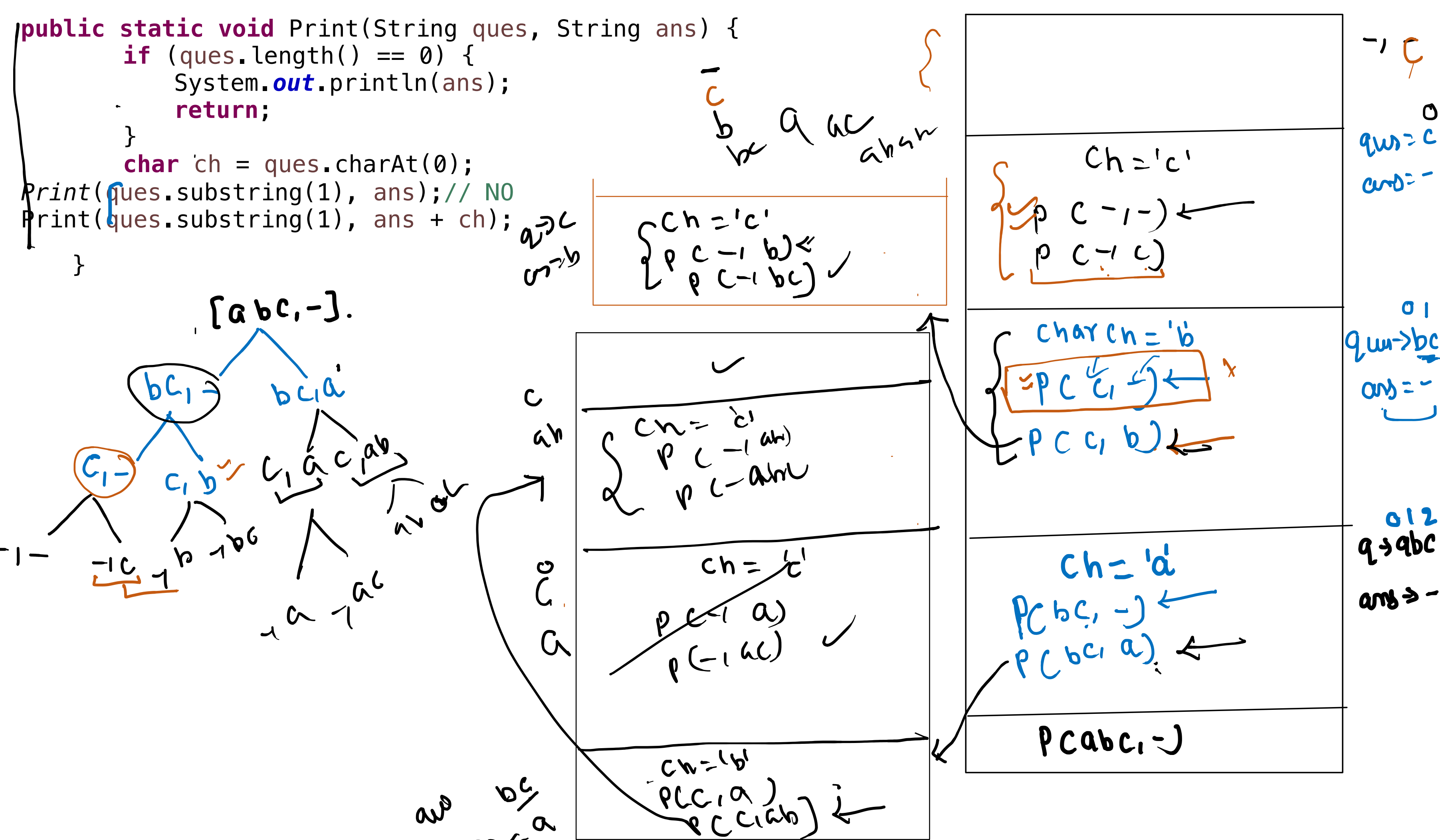
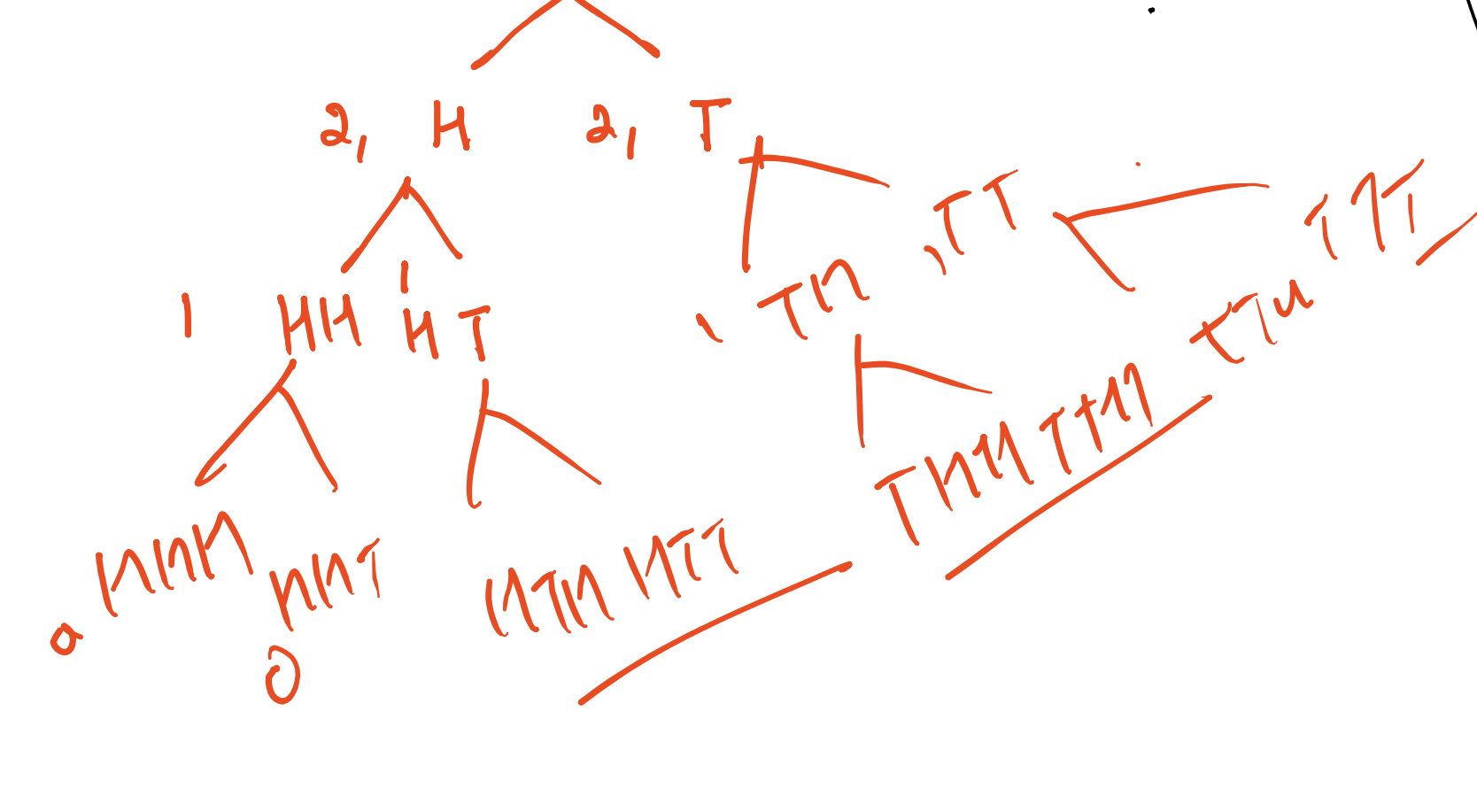


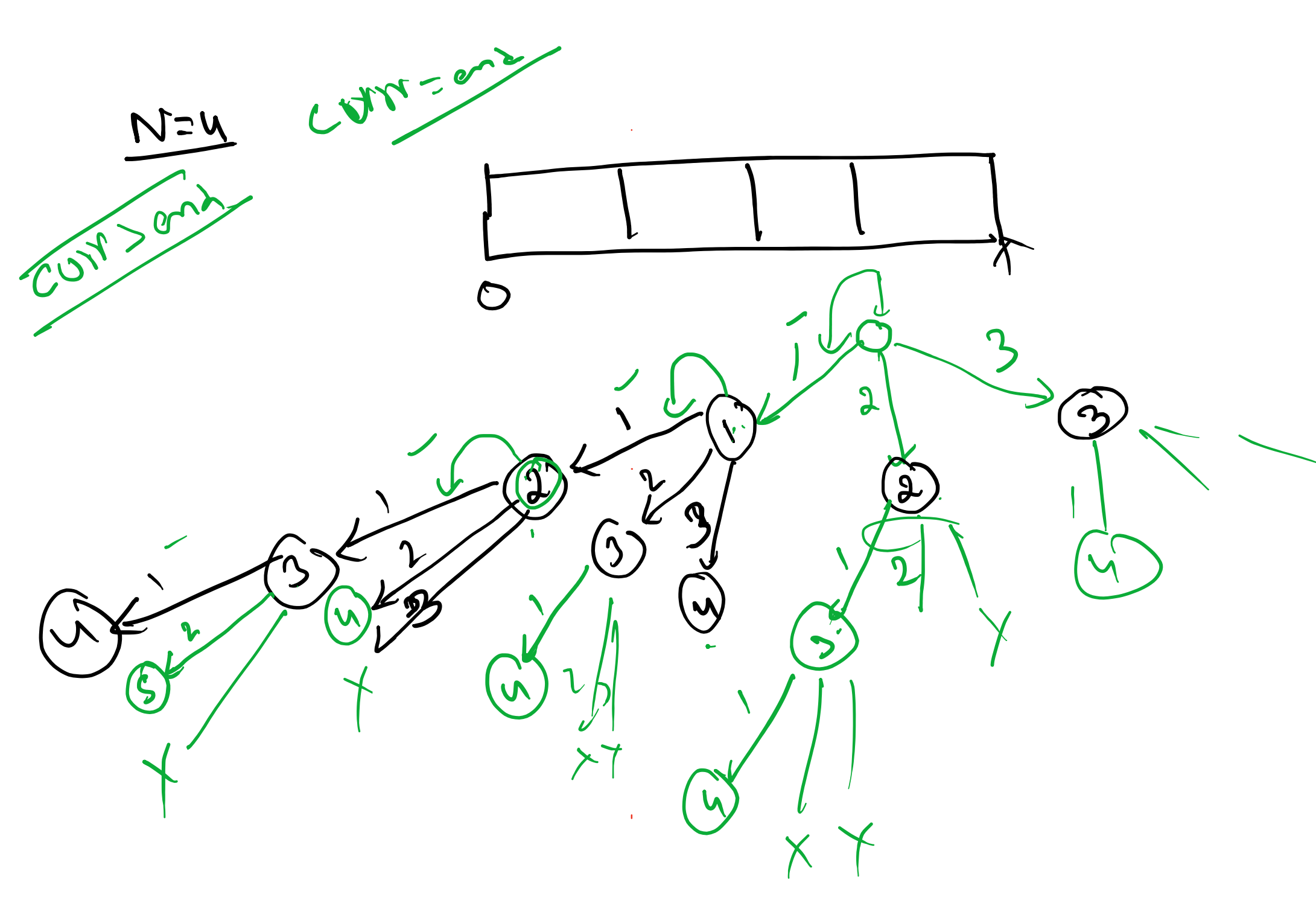
[illegible][illegible]

n-1 and:



A handwritten tree diagram illustrating the Huffman coding process. The root node is 'A'. It branches into '2, B' and '2, T'. '2, B' branches into '1, MM' and '1, NT'. '1, MM' branches into 'MM' and '2, MT'. '1, NT' branches into 'MT' and 'TT'. '2, MT' branches into 'TMT' and 'TMT'. 'TT' branches into 'TMT' and 'TMT'. The final tree structure shows the binary representation of the characters: MM (01), MT (00), TMT (10), and TT (11).

Handwritten diagram illustrating a binary tree structure for a Huffman tree. The root node is labeled  $N=3, -$ . The left child is labeled  $2H$  (circled). The right child is labeled  $2T$ . The  $2H$  node has two children:  $X$  and  $HT$ . The  $2T$  node has two children:  $T(1)$  (circled) and  $T$  (crossed out). The  $T(1)$  node has two children:  $X$  and  $THT$ . The  $T$  node has one child:  $THT$ . The  $HT$  node has two children:  $HTH$  and  $HTT$ .



dice = 3  $\{ (7, 8, 9) \}$

1111 1111  
112

$$\begin{array}{r} 1111 \\ 112 \\ 121 \\ 12 \\ 211 \\ 22 \\ 31 \end{array}$$

String ans) {

$n = 3, \text{ call} = 0$

1111111  
112  
121  
13  
211  
22  
31

Handwritten notes on a grid showing the calculation of the probability of a sum of 3 from two dice rolls. The notes are organized into three rows, each representing a different sum (3, 4, 5). Each row shows the possible outcomes (e.g., (1,1), (1,2), (2,1)) and the total number of outcomes (N=3). The probability is calculated as P(sum, 3, 111) for sum 3, P(sum, 3, 12) for sum 4, and P(sum, 3, 12) for sum 5. The final result is P(3, 01-).



if  $\text{Oprn} == n$  then

if (open > n || closed > q) {

مجلس

```

    if (open > n || closed > open) {
        return;
    }

    Parentheses(n, open + 1, closed, ans + "(");
    Parentheses(n, open, closed + 1, ans + ")");
}

```

