Problem: Given string $S = \{x_1, x_2, \dots x_n\}$, how many permutations of this string, including this one, will have the same binary decision tree?

Approach 1: (Using the tree only)

Algorithm:

1. Draw the tree and label the n nodes $N_1, N_2, \dots, N_n$.
2. For each node N_i define $Count(N_i) = aCb$ where a is the number of nodes on BOTH of N_i's branches and b is the number of nodes on the left branch only. (For b, It makes no difference which branch is chosen, so you can count nodes on the right branch instead to determine b. b itself will likely change, but aCb will remain the same.)
3. Total number of permutation of S that generate the same tree as S is

$$\prod_{i=1}^{n} Count(N_i)$$

NOTE: It's worthy of note that if a node does not branch into two branches, the count of that node will be 1. So all such nodes can be ignored in the performance of the algorithm.

Approach 2: (Using the string only)

Algorithm:

1. For a given string S, define $Count(S) = aCb$ where a is 1 less than the number of elements in S and b is the number of elements after the first element that are less than the first element. (b could also be defined as the number of elements after the first greater than or equal to the first.)
2. From the given string S, create two strings as follows:
   a. Delete the first element and call this string S'
   b. From S', delete all elements greater than or equal to the first element of S and call this new string S_{left}.
   c. From S', delete all elements less than the first element of S and call this new string S_{right}.
3. Repeat steps 1 and 2 for the strings S_{left} and S_{right}, generating two more counts and potentially 4 more strings.
4. Continue creating counts for substrings created and breaking them into smaller substrings until you get all substrings having 0, 1, or 2 elements. Since the count of such strings is 1, there is no reason to continue.
5. Multiply all counts together to get the total number of permutations of the string that will generate the same tree.