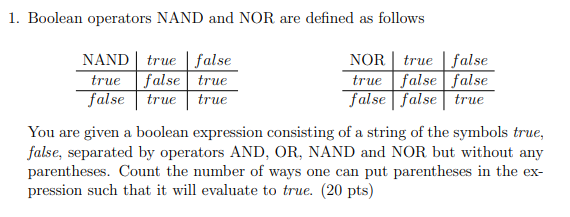
ASSIGNMENT4 QUESTION1



**Answer:**

We have a Boolean expression consisting of a string [i:j], and for every part of string, we want to know the numbers of way that it be true and false.

Let T[i , j] be the number of ways of parenthesizing between the string [i : j] evaluates to TRUE. F[i , j] be the number of ways of parenthesizing between between the string [i : j] evaluates to False. Include I j. (I j)

Then the total[i , j] = T[i, j] + F[i , j]

T[i , j] and F[i , j] can be expressed by the following recursive formulations:

And we have this base case:

T[i , i] = 1 if S[i] = True

T[i , i] = 0 if S[i] = False

F[i , i] = 0 if S[i] = True

F[i , i] = 1 if S[i] = False

The final answer will be given by T[1,n] since we use all string [1:n]. and we want the string evaluate to true.

The time complexity for this question should be since we have sub-problems and to solve each problems .