# Question 1

## Setup

Let’s assume there are **n** symbols, and **m** operations **(m = n – 1)**;

## Subproblem

Subproblem is how many ways to make true expression from to ;

And we also calculate how many ways to make false in the same way.

Mark as the number of ways of making the expression **true** from to ;

Mark as the number of ways of making the expression **false** from to .

We split the strings by an operator **m** so we only focus the left side and right side of the operator then do the recursion.

## Recursion:

Solve both subproblem in parallel.

## Base case

## Final solution

Do the , then after recursion function we can get the answer.

## Complexity

The complexity is , there are n symbols and form at most outcome and there are n operations, therefore we get .