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 i_{th} to j_{th};

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

Mark $\boldsymbol{T}(\boldsymbol{i},\boldsymbol{j})$ as the number of ways of making the expression **true** from i_{th} to j_{th};

Mark $\boldsymbol{F}(\boldsymbol{i},\boldsymbol{j})$ as the number of ways of making the expression **false** from i_{th} to j_{th}.

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.

$$T(i,j) = \sum_{m=i}^{j-1} TSub(i,j,m)$$

$$F(i,j) = \sum_{m=i}^{j-1} FSub(i,j,m)$$

$TSub(i,j,m)$

$$= \begin{cases} T(i,m) \bullet T(m+1,j) \ \ if\ m = AND \\ T(i,m) \bullet F(m+1,j) + F(i,m) \bullet T(m+1,j)\ if\ m = OR \\ T(i,m) \bullet F(m+1,j) + F(i,m) \bullet T(m+1,j) + F(i,m) \bullet F(m+1,j)\ if\ m = NAND \\ F(i,m) \bullet F(m+1,j) \ \ if\ m = NOR \end{cases}$$

$FSub(i,j,m)$

$$= \begin{cases} T(i,m) \bullet F(m+1,j) + F(i,m) \bullet T(m+1,j) + F(i,m) \bullet F(m+1,j) \ \ if\ m = AND \\ F(i,m) \bullet F(m+1,j)\ \ if\ m = OR \\ T(i,m) \bullet T(m+1,j)\ if\ m = NAND \\ T(i,m) \bullet F(m+1,j) + F(i,m) \bullet T(m+1,j) + T(i,m) \bullet T(m+1,j)\ \ if\ m = NOR \end{cases}$$

Base case

$T(i,i) = 1\ if\ i\ is\ true$
$T(i,i) = 0\ if\ i\ is\ false$
$F(i,i) = 1\ if\ i\ is\ false$
$F(i,i) = 0\ if\ i\ is\ true$

Final solution

$$T(i,j) = \sum_{m=i}^{j-1} TSub(i,j,m) : i \leq n, j \leq n$$

Do the $T(1,n)$, then after recursion function we can get the answer.

Complexity

The complexity is $O(n^3)$, there are n symbols and form at most n^2 outcome and there are n operations, therefore we get $O(n^3)$.