z5158334
Yuanzhi Shang

Question 1

For this question, Dynamic programming method could be applied. To solve the problem for the whole string, it needs to be first considered that subproblem. That is, the optimal solution between i^{th} and j^{th} true or false in the Boolean expression. Assume $T(i,j)$ would be the number of ways to placement bracket that can make the result to be true between i,j and $F(i,j)$ would be the number of ways that can make it to be false. Both $T(i,j)$ and $F(i,j)$ would be the **optimal solution** for substring (i,j)

This Boolean expression could be split by specific operator and form two subproblems. It can be represented by following equation.

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

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

Where (i,j) could be a substring and $TS\ and\ FS$ represent that string (i,j) split by operator m then form two substrings.
The value of $FS()$ and $TS()$ would be depend on the specific type of operator.

1. The value of $TS(i,m,j)$
   a) "AND" : $T(i,m) \times T(m+1,j)$ (the only way to get true is "true & true")
   b) "OR": $T(i,m) \times F(m+1,j) + T(j,m) \times T(m+1,j) + F(i,m) \times T(m+1,j)$(three situations that we can get true)
   c) "NAND": $T(i,m) \times F(m+1,j) + F(j,m) \times F(m+1,j) + F(i,m) \times T(m+1,j)$
   d) "NOR": $F(i,m) \times F(m+1,j)$
2. The value of $FS(i,m,j)$ is opposite
   a) "AND": $T(i,m) \times F(m+1,j) + F(j,m) \times F(m+1,j) + F(i,m) \times T(m+1,j)$
   b) "OR": $F(i,m) \times F(m+1,j)$
   c) "NAND": $T(i,m) \times T(m+1,j)$
   d) "NOR": $T(i,m) \times F(m+1,j) + T(j,m) \times T(m+1,j) + F(i,m) \times T(m+1,j)$

Basically, for example, if we split string by any operators, the optimal solution is the combination of optimal solution of the two parts that we create. The algorithm run recursively and solve the base situation which is string (i,i) and then return to solve the bigger subproblems.