9101 Assignment 3
Haojin Guo
z5216214

Q1

1. Boolean operators NAND and NOR are defined as follows

NAND	$true$	$false$
$true$	$false$	$true$
$false$	$true$	$true$

NOR	$true$	$false$
$true$	$false$	$false$
$false$	$false$	$true$

You are given a boolean expression consisting of a string of the symbols *true*, *false*, separated by operators AND, OR, NAND and NOR but without any parentheses. Count the number of ways one can put parentheses in the expression such that it will evaluate to *true*. (20 pts)

Solution:

1) Let $T(i,j)$ represents the number of ways to parenthesize the symbols between i and j (both inclusive) such that the subexpression between i and j evaluates to True.
   Let $F(i,j)$ represents the number of ways to parenthesize the symbols between i and j (both inclusive) such that the subexpression between i and j evaluates to False.
   Symbol:
   $T = true$,
   $F = false$,
   $\& \rightarrow AND$,
   $| \rightarrow OR$
2) Base case:
   a) $T(i,i) = 1,\ F(i,i) = 0,\quad if\ symbol[i] = T$,
   b) $T(i,i) = 0,\ F(i,i) = 1,\quad if\ symbol[i] = F$,
3) Recursion process:
   a) For $T(i,j)$, assuming that we have solved subproblems for $T(i,k)$ and $T(k+1,j)$ where $k \in [i,j]$
   b) For $F(i,j)$, assuming that we have solved subproblems for $F(i,k)$ and $F(k+1,j)$ where $k \in [i,j]$
   c) Then, for the specific state transition equation, there are 4 cases in this question.
      i. CASE 1: operation = AND
         1. $T(i,j) = \sum_{k=i}^{j-1} T(i,k) * T(k+1,j)$
         2. $F(i,j) = \sum_{k=i}^{j-1} F(i,k) * F(k+1,j) + F(i,k) * T(k+1,j) + T(i,k) * F(k+1,j)$

      ii. CASE 2: operation = OR
         1. $T(i,j) = \sum_{k=i}^{j-1} T(i,k) * T(k+1,j) + T(i,k) * F(k+1,j) + F(i,k) * T(k+1,j)$
         2. $F(i,j) = \sum_{k=i}^{j-1} F(i,k) * F(k+1,j)$

iii. CASE 3: operation = NAND

1. $T(i,j) = \sum_{k=i}^{j-1} T(i,k) * F(k+1,j) + F(i,k) * T(k+1,j) + F(i,k) * F(k+1,j)$
2. $F(i,j) = \sum_{k=i}^{j-1} T(i,k) * T(k+1,j)$

iv. CASE 4: operation = NOR

1. $T(i,j) = \sum_{k=i}^{j-1} F(i,k) * F(k+1,j)$
2. $F(i,j) = \sum_{k=i}^{j-1} T(i,k) * T(k+1,j) + F(i,k) * T(k+1,j) + T(i,k) * F(k+1,j)$

To summarize the above：

$$T(i,j) = \sum_{k=i}^{j-1} \begin{cases} T(i,k) * T(k+1,j) & if\ operation\ = AND \\ T(i,k) * T(k+1,j) + T(i,k) * F(k+1,j) + F(i,k) * T(k+1,j) & if\ operation\ = OR \\ T(i,k) * F(k+1,j) + F(i,k) * T(k+1,j) + F(i,k) * F(k+1,j) & if\ operation = NAND \\ F(i,k) * F(k+1,j) & if\ operation\ = NOR \end{cases}$$

$$F(i,j) = \sum_{k=i}^{j-1} \begin{cases} F(i,k) * F(k+1,j) + F(i,k) * T(k+1,j) + T(i,k) * F(k+1,j) & if\ operation\ = AND \\ F(i,k) * F(k+1,j) & if\ operation\ = OR \\ T(i,k) * T(k+1,j) & if\ operation = NAND \\ T(i,k) * T(k+1,j) + F(i,k) * T(k+1,j) + T(i,k) * F(k+1,j) & if\ operation\ = NOR \end{cases}$$

For example, for the expression,

$$T\ \ |\ \ F\ \ NAND\ \ T\ \ \&\ T\ \ ,$$
$$1\ -2\ \ ---\ 3\ \ --\ \ 4$$

Based on the equations above, we have,

T(1,2) = T(1,1)*T(2,2) + T(1,1)*F(2,2)+T(1,1)+T(2,2) = 1

T(2,3) = F(2,2)*F(3,3) + F(2,2)*T(3,3)+T(2,2)*F(3,3) = 1

T(3,4) = T(3,3)*T(4,4) =1

(k=1,k=2,,)

T(1,3) = T(1,1)*F(2,3)+T(1,1)*T(2,3)+F(1,1)*T(2,3)

+ T(1,2)*F(3,3)+F(1,2)*F(3,3)+F(1,2)*T(3,3) = 1

(k=2,3)

T(2,4) =T(2,2)*F(3,4) +F(2,2)*T(3,4)+F(2,2)*F(3,4)

+T(2,3)*T(4,4) = 2

(k=1,2,3)

T(1,4) = T(1,1)*T*(2,4) +T(1,2)*T(3,4)+T(1,3)*T(4,4) = 4

4) We can solve this problem by filling two tables in the top to the bottom manner just like below and get the answer at the last blank $\boldsymbol{T}(1, n)$ on the True table.

$$number\ of\ ways = T(1, n)$$

T(i, j)	**1**	**2**	**3**	**4**
1	1	1	1	4
2		0	1	2
3			1	1
4				1

F(i, j)	**1**	**2**	**3**	**4**
1	0			
2		1		
3			0	
4				0

5) Time complexity: O (n^3)