HOMEWORK 4 – Q1

MINGLANG XIE

z5228006

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:

Setup: Let there be n symbols, and $n-1$ operations between them.

Subproblems: Contiguous substrings of that boolean expression between i^{th} and j^{th} true/false symbols and solve subproblems which involve counting both the number of placement of the brackets such that the subexpression evaluates true, and the number of ways it evaluates to false.

Build-up order:

Solve the subproblems in the order $T(i,0), T(i,1) \dots T(i,n)$.

(Bottom-up approach)

Recursion:

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.

$$T(i,j) = \sum_{k=i}^{j-1} \begin{cases} T(i,k) * T(k+1,j) \\ \qquad if\ operator\ k\ is\ 'AND', \\ T(i,k) * F(k+1,j) + T(i,k) * T(k+1,j) + F(i,k) * T(k+1,j) \\ \qquad if\ operator\ k\ is\ 'OR', \\ T(i,k) * F(k+1,j) + F(i,k) * F(k+1,j) + F(i,k) * T(k+1,j) \\ \qquad if\ operator\ k\ is\ 'NAND', \\ F(i,k) * F(k+1,j) \\ \qquad if\ operator\ k\ is\ 'NOR'. \end{cases}$$

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.

$$F(i,j) = \sum_{k=i}^{j-1} \begin{cases} T(i,k) * F(k+1,j) + F(i,k) * F(k+1,j) + F(i,k) * T(k+1,j) \\ \qquad if\ operator\ k\ is\ 'AND', \\ F(i,k) * F(k+1,j) \\ \qquad if\ operator\ k\ is\ 'OR', \\ T(i,k) * T(k+1,j) \\ \qquad if\ operator\ k\ is\ 'NAND', \\ T(i,k) * F(k+1,j) + T(i,k) * T(k+1,j) + F(i,k) * T(k+1,j) \\ \qquad if\ operator\ k\ is\ 'NOR'. \end{cases}$$

Base case:

$T(i,i) = 1\ if\ symbol[i]\ is\ True$
$T(i,i) = 0\ if\ symbol[i]\ is\ False$

$F(i,i) = 1\ if\ symbol[i]\ is\ False$
$F(i,i) = 0\ if\ symbol[i]\ is\ True$

Final solution:

$$The\ final\ solution\ is\ given\ by\ T(1,n)$$

Time complexity:

The time complexity is $O(n^3)$. There are n^2 different ranges that i and j could cover, and each needs the evaluations of $T(i,j)$ or $F(i,j)$ at up to n different time.