TIC TAC TOE ALGORITHM

```
board = {
    1: ' ', 2: ' ', 3: ' ',
    4: ' ', 5: ' ', 6: ' ',
    7: ' ', 8: ' ', 9: ' '
}

def printBoard(board):
    print(board[1] + ' | ' + board[2] + ' | ' + board[3])
    print('--+---+--')
    print(board[4] + ' | ' + board[5] + ' | ' + board[6])
    print('--+---+--')
    print(board[7] + ' | ' + board[8] + ' | ' + board[9])
    print("\n")

def spaceFree(pos):
    return board[pos] == ' '

def checkWin():
    win_conditions = [
        (1, 2, 3), (4, 5, 6), (7, 8, 9),  # Rows
        (1, 4, 7), (2, 5, 8), (3, 6, 9),  # Columns
        (1, 5, 9), (3, 5, 7)              # Diagonals
    ]
    for a, b, c in win_conditions:
        if board[a] == board[b] == board[c] and board[a] != ' ':
            return True
```

```
    return False

def checkMoveForWin(move):
    win_conditions = [
        (1, 2, 3), (4, 5, 6), (7, 8, 9),
        (1, 4, 7), (2, 5, 8), (3, 6, 9),
        (1, 5, 9), (3, 5, 7)
    ]
    for a, b, c in win_conditions:
        if board[a] == board[b] == move and board[a] != ' ':
            return True
    return False

def checkDraw():
    return all(board[key] != ' ' for key in board.keys())

def insertLetter(letter, position):
    if spaceFree(position):
        board[position] = letter
        printBoard(board)

        if checkDraw():
            print("Draw!")
            return
        elif checkWin():
            if letter == 'X':
                print("Bot wins!")
            else:
                print("You win!")
            return

    else:
```

```
        print("Position taken, please pick a different position.")
        position = int(input("Enter new position for O: "))
        insertLetter(letter, position)

player = 'O'
bot = 'X'

def playerMove():
    position = int(input('Enter position for O: '))
    insertLetter(player, position)

def compMove():
    bestScore = -1000
    bestMove = 0
    for key in board.keys():
        if board[key] == ' ':
            board[key] = bot
            score = minimax(board, False)
            board[key] = ' '
            if score > bestScore:
                bestScore = score
                bestMove = key

    insertLetter(bot, bestMove)

def minimax(board, isMaximizing):
    if checkMoveForWin(bot):
        return 1
    elif checkMoveForWin(player):
        return -1
    elif checkDraw():
        return 0
```

```
    if isMaximizing:
        bestScore = -1000
        for key in board.keys():
            if board[key] == ' ':
                board[key] = bot
                score = minimax(board, False)
                board[key] = ' '
                bestScore = max(score, bestScore)
        return bestScore
    else:
        bestScore = 1000
        for key in board.keys():
            if board[key] == ' ':
                board[key] = player
                score = minimax(board, True)
                board[key] = ' '
                bestScore = min(score, bestScore)
        return bestScore

while not checkWin() and not checkDraw():
    compMove()
    if checkWin() or checkDraw():
        break
    playerMove()

print("Tanush Prajwal S")
print("1BM22CS304")
```

OUTPUT :

```
X |   |
--+---+--
  |   |
--+---+--
  |   |


Enter position for O: 1
Position taken, please pick a different position.
Enter new position for O: 2
X | O |
--+---+--
  |   |
--+---+--
  |   |


X | O |
--+---+--
X |   |
--+---+--
  |   |


Enter position for O: 7
X | O |
--+---+--
X |   |
--+---+--
O |   |
```

```
X | O | X
--+---+--
X |   |
--+---+--
O |   |


Enter position for O: 6
X | O | X
--+---+--
X |   | O
--+---+--
O |   |


X | O | X
--+---+--
X | X | O
--+---+--
O |   |


Enter position for O: 9
X | O | X
--+---+--
X | X | O
--+---+--
O |   | O


X | O | X
--+---+--
X | X | O
--+---+--
O | X | O


Draw!
Tanush Prajwal S
1BM22CS304


...Program finished with exit code 0
Press ENTER to exit console.
```