64. **Determine if a `9 x 9` Sudoku board is valid. Only the filled cells need to be validated according to the following rules:**

1. **Each row must contain the digits `1-9` without repetition.**
2. **Each column must contain the digits `1-9` without repetition.**
3. **Each of the nine `3 x 3` sub-boxes of the grid must contain the digits `1-9` without repetition.**

Code:

```
def isValidSudoku(board):
    for row in board:
        seen = set()
        for num in row:
            if num != '.':
                if num in seen:
                    return False
                seen.add(num)
    for col in range(9):
        seen = set()
        for row in range(9):
            num = board[row][col]
            if num != '.':
                if num in seen:
                    return False
                seen.add(num)
    for i in range(3):
        for j in range(3):
            seen = set()
            for m in range(3):
                for n in range(3):
                    num = board[3*i + m][3*j + n]
                    if num != '.':
                        if num in seen:
                            return False
                        seen.add(num)
    return True
```

```
board1 = [
    ["5","3",".",".","7",".",".",".","."],
    ["6",".",".","1","9","5",".",".","."],
    [".","9","8",".",".",".",".","6","."],
    ["8",".",".",".","6",".",".",".","3"],
    ["4",".",".","8",".","3",".",".","1"],
    ["7",".",".",".","2",".",".",".","6"],
    [".","6",".",".",".",".","2","8","."],
    [".",".",".","4","1","9",".",".","5"],
    [".",".",".",".","8",".",".","7","9"]
]
board2 = [
    ["8","3",".",".","7",".",".",".","."],
    ["6",".",".","1","9","5",".",".","."],
    [".","9","8",".",".",".",".","6","."],
    ["8",".",".",".","6",".",".",".","3"],
    ["4",".",".","8",".","3",".",".","1"],
    ["7",".",".",".","2",".",".",".","6"],
    [".","6",".",".",".",".","2","8","."],
    [".",".",".","4","1","9",".",".","5"],
    [".",".",".",".","8",".",".","7","9"]
]
print(isValidSudoku(board1))
print(isValidSudoku(board2))
```

Output:

```
True
False
```

Time Complexity:

- T(n)= O(9^m+n)