

Valid sudoku - using bitmask

class Solution:

def is_valid_sudoku(self, board: list[list[str]]) -> bool:

rows = [0] * 9

cols = [0] * 9

squares = [0] * 9

for r in range(9):

for c in range(9):

if board[r][c] == ".":

continue

val = int(board[r][c]) - 1

if (1 <= val) & rows[r]:

return False

if (1 <= val) & cols[c]:

return False

if (1 <= val) & squares[(r//3)*3 + (c//3)]:

return False

rows[r] 1 = (1 <= val)

cols[c] 1 = (1 <= val)

squares[(r//3)*3 + (c//3)] 1 = (1 <= val)

return True

```
print(Solution().is_valid_sudoku(board =
[[1, 2, 3, ., ., ., ., ., .],
[4, 5, ., ., ., ., ., ., .],
[. 8, 3, ., ., ., ., ., 3],
[5, ., 6, ., ., ., 4, ., .],
[. 4, 8, ., 3, ., ., ., 5],
[7, ., 2, ., ., ., ., 6, .],
[. ., ., ., ., 2, ., ., .],
[. ., ., 4, 1, 9, ., ., 8],
[. ., ., 8, ., ., 7, 9, .]]))
```

This exercise is using bitmask algorithm and is running in approximately 431 steps.

Step 1:

rows → [0, 0, 0, 0, 0, 0, 0, 0, 0]

cols $\rightarrow [0, 0, 0, 0, 0, 0, 0, 0, 0]$
 squares $\rightarrow [0, 0, 0, 0, 0, 0, 0, 0, 0]$

Info: recap bitmask operators:

formula:

- left shift (\ll): $a \ll n = a * 2^n$

for exp: $a=5, n=3$

$$5 \ll 3 = 5 * 2^3 = 40$$

$$100111 \Rightarrow 32 + 0 + 0 + 4 + 2 + 1 = 39 \ll 1 \Rightarrow 1001110 \Rightarrow 78$$

$2^5 - - 2^2 2^1 2^0$
 $2^6 - - 2^3 2^2 2^1 2^0$
 $\Rightarrow 78$

- Bitwise OR ($|$): $(a | b)_i = a_i + b_i - (a_i \times b_i)$

for exp: $3 = 1001$

$$14 = 1110$$

$$15 = 1111$$