0036. 有效的数独

ITCharge 大约 2 分钟

- 标签：数组、哈希表、矩阵
- 难度：中等

题目链接

- 0036. 有效的数独 - 力扣

题目大意

描述：给定一个数独，用 `9 * 9` 的二维字符数组 `board` 来表示，其中，未填入的空白用 "." 代替。

要求：判断该数独是否是一个有效的数独。

说明：

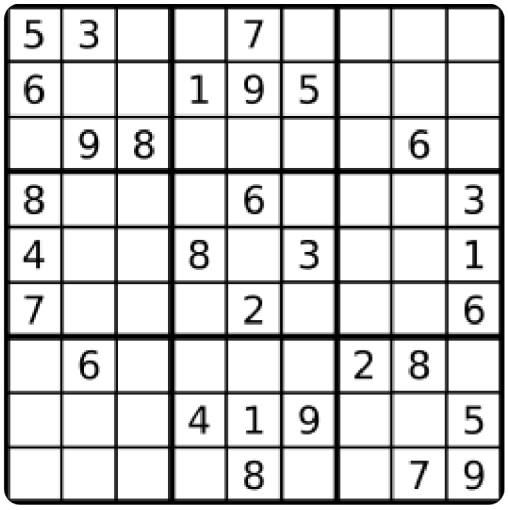
- 一个有效的数独（部分已被填充）不一定是可解的。
- 只需要根据以上规则，验证已经填入的数字是否有效即可。
- 空白格用 `'.'` 表示。

一个有效的数独需满足：

1. 数字 `1-9` 在每一行只能出现一次。
2. 数字 `1-9` 在每一列只能出现一次。
3. 数字 `1-9` 在每一个以粗实线分隔的 `3 * 3` 宫内只能出现一次。（请参考示例图）

示例：

- 示例 1：



```py
输入：board =
[["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    "]]
输出：True
```

解题思路

思路 1：哈希表

判断数独有效，需要分别看每一行、每一列、每一个 `3 * 3` 的小方格是否出现了重复数字，如果都没有出现重复数字就是一个有效的数独，如果出现了重复数字则不是有效的数独。

- 用 `3` 个 `9 * 9` 的数组分别来表示该数字是否在所在的行，所在的列，所在的方格出现过。其中方格角标的计算用 `box[(i / 3) * 3 + (j / 3)][n]` 来表示。
- 双重循环遍历数独矩阵。如果对应位置上的数字如果已经在在所在的行 / 列 / 方格出现过，则返回 `False` 。
- 遍历完没有重复出现，则返回 `Ture` 。

思路 1：代码

```py
class Solution:
    def isValidSudoku(self, board: List[List[str]]) -> bool:
        rows_map = [dict() for _ in range(9)]
        cols_map = [dict() for _ in range(9)]
        boxes_map = [dict() for _ in range(9)]

        for i in range(9):
            for j in range(9):
                if board[i][j] == '.':
                    continue
                num = int(board[i][j])
                box_index = (i // 3) * 3 + j // 3
                row_num = rows_map[i].get(num, 0)
                col_num = cols_map[j].get(num, 0)
                box_num = boxes_map[box_index].get(num, 0)
                if row_num > 0 or col_num > 0 or box_num > 0:
                    return False
                rows_map[i][num]
                cols_map[j][num] = 1
                boxes_map[box_index][num] = 1

        return True
```

思路 1：复杂度分析

- **时间复杂度**：$O(1)$。数独总共 81 个单元格，对每个单元格遍历一次，可以看做是常数级的时间复杂度。
- **空间复杂度**：$O(1)$。使用 81 个单位空间，可以看做是常数级的空间复杂度。