class Solution {

public:

bool isValidSudoku(vector<vector<char>>& board) {

vector <vector<int>> num[10];

// check by row and box

for (int row = 0, col, size, n; row < 9; ++row)

{

for (col = 0; col < 9; ++col)

{

if (board[row][col] != '.')

{

n = board[row][col] - '0';

size = num[n].size();

if (size > 0 && (num[n][size - 1][1] == row

|| (num[n][size - 1][1] / 3 == row / 3

&& num[n][size - 1][0] / 3 == col / 3)

|| (size > 1 && num[n][size - 2][1] / 3 == row / 3

&& num[n][size - 2][0] / 3 == col / 3)))

return false;

num[n].push\_back(vector <int>{ col, row });

}

}

}

// check by col

for (int i = 1; i <= 9; ++i)

sort(num[i].begin(), num[i].end());

for (int i = 1, j, len; i <= 9; ++i)

{

for (j = 0, len = num[i].size() - 1; j < len; ++j)

if (num[i][j][0] == num[i][j + 1][0])

return false;

}

return true;

}

};