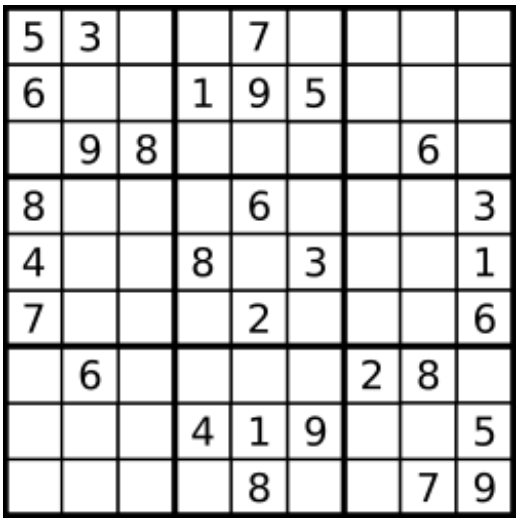
Valid Sudoku

Determine if a Sudoku is valid, according to: Sudoku Puzzles - The Rules.

The Sudoku board could be partially filled, where empty cells are filled with the character `'.'`.



A partially filled sudoku which is valid.

Note:
A valid Sudoku board (partially filled) is not necessarily solvable. Only the filled cells need to be validated.

Solution 1

Three flags are used to check whether a number appear.

used1: check each row

used2: check each column

used3: check each sub-boxes

```
class Solution
{
public:
    bool isValidSudoku(vector<vector<char> > &board)
    {
        int used1[9][9] = {0}, used2[9][9] = {0}, used3[9][9] = {0};

        for(int i = 0; i < board.size(); ++ i)
            for(int j = 0; j < board[i].size(); ++ j)
                if(board[i][j] != '.')
                {
                    int num = board[i][j] - '0' - 1, k = i / 3 * 3 + j / 3;
                    if(used1[i][num] || used2[j][num] || used3[k][num])
                        return false;
                    used1[i][num] = used2[j][num] = used3[k][num] = 1;
                }

        return true;
    }
};
```

written by makuiyu original link here

Solution 2

```
public class Solution {
public boolean isValidSudoku(char[][] board) {
    for (int i=0; i<9; i++) {
        if (!isParticallyValid(board,i,0,i,8)) return false;
        if (!isParticallyValid(board,0,i,8,i)) return false;
    }
    for (int i=0;i<3;i++){
        for(int j=0;j<3;j++){
            if (!isParticallyValid(board,i*3,j*3,i*3+2,j*3+2)) return false;
        }
    }
    return true;
}
private boolean isParticallyValid(char[][] board, int x1, int y1,int x2,int y2){
    Set singleSet = new HashSet();
    for (int i= x1; i<=x2; i++){
        for (int j=y1;j<=y2; j++){
            if (board[i][j]!='.') if(!singleSet.add(board[i][j])) return false;
        }
    }
    return true;
}
```

}

Each time send the coordinates to check if the board is partially valid.

written by bigwolfandtiger original link here

Solution 3

```
 bool isValidSudoku(vector<vector<char>>& board) {
  vector<short> col(9, 0);
  vector<short> block(9, 0);
  vector<short> row(9, 0);
  for (int i = 0; i < 9; i++)
   for (int j = 0; j < 9; j++) {
       if (board[i][j] != '.') {
           int idx = 1 << (board[i][j] - '0');
           if (row[i] & idx || col[j] & idx || block[i/3 * 3 + j / 3] & idx)
              return false;
          row[i] |= idx;
          col[j] |= idx;
          block[i/3 * 3 + j/3] |= idx;
       }
   }
   return true;
}
```

written by lchen77 original link here