```c
#define SUB_SIZE                            (3)
#define TOTAL_DIGITS                        (9)


bool isValidSudoku(char** board, int boardSize, int* boardColSize){
    int sub_board_size;
    int sub_board_column_size;
    int total_sub_board;
    int** column_digit;
    int* alloc_column_digit;
    int** row_digit;
    int* alloc_row_digit;
    int** sub_digit;
    int* alloc_sub_digit;
    int row_index;
    int column_index;
    int sub_index;

    sub_board_size = boardSize / SUB_SIZE;
    sub_board_column_size = *boardColSize / SUB_SIZE;
    total_sub_board = sub_board_size * sub_board_column_size;

    column_digit = (int**)malloc(sizeof(int*)*(*boardColSize));
    memset(column_digit, 0x0, sizeof(int*)*(*boardColSize));
    row_digit = (int**)malloc(sizeof(int*)*boardSize);
    memset(row_digit, 0x0, sizeof(int*)*boardSize);
    sub_digit = (int**)malloc(sizeof(int*)*total_sub_board);
    memset(sub_digit, 0x0, sizeof(int*)*total_sub_board);

    alloc_column_digit = (int*)malloc(sizeof(int)*(*boardColSize)*TOTAL_DIGITS);
    memset(alloc_column_digit, 0x0, sizeof(int)*(*boardColSize)*TOTAL_DIGITS);
    alloc_row_digit = (int*)malloc(sizeof(int)*boardSize*TOTAL_DIGITS);
    memset(alloc_row_digit, 0x0, sizeof(int)*boardSize*TOTAL_DIGITS);
    alloc_sub_digit = (int*)malloc(sizeof(int)*total_sub_board*TOTAL_DIGITS);
    memset(alloc_sub_digit, 0x0, sizeof(int)*total_sub_board*TOTAL_DIGITS);

    for(row_index = 0; row_index < boardSize; row_index++)
    {
        for(column_index = 0; column_index < *boardColSize; column_index++)
        {
            if(board[row_index][column_index] != '.')
            {
                if(column_digit[column_index] == NULL)
                {
                   column_digit[column_index] = &alloc_column_digit[column_index*TOTAL_DIGITS];
                }

                if(column_digit[column_index][board[row_index][column_index]-'1'] >= 1)
                {
                    return false;
                }else
                {
                    column_digit[column_index][board[row_index][column_index]-'1']++;
                }

                if(row_digit[row_index] == NULL)
                {
                   row_digit[row_index] = &alloc_row_digit[row_index*TOTAL_DIGITS];
                }

                if(row_digit[row_index][board[row_index][column_index]-'1'] >= 1)
                {
                    return false;
                }else
                {
                    row_digit[row_index][board[row_index][column_index]-'1']++;
                }

                sub_index = (column_index/SUB_SIZE)*sub_board_size + (row_index/SUB_SIZE);

                if(sub_digit[sub_index] == NULL)
                {
                   sub_digit[sub_index] = &alloc_sub_digit[sub_index*TOTAL_DIGITS];
                }

                if(sub_digit[sub_index][board[row_index][column_index]-'1'] >= 1)
                {
                    return false;
                }else
                {
                    sub_digit[sub_index][board[row_index][column_index]-'1']++;
                }
            }
        }
    }

    return true;
}
```