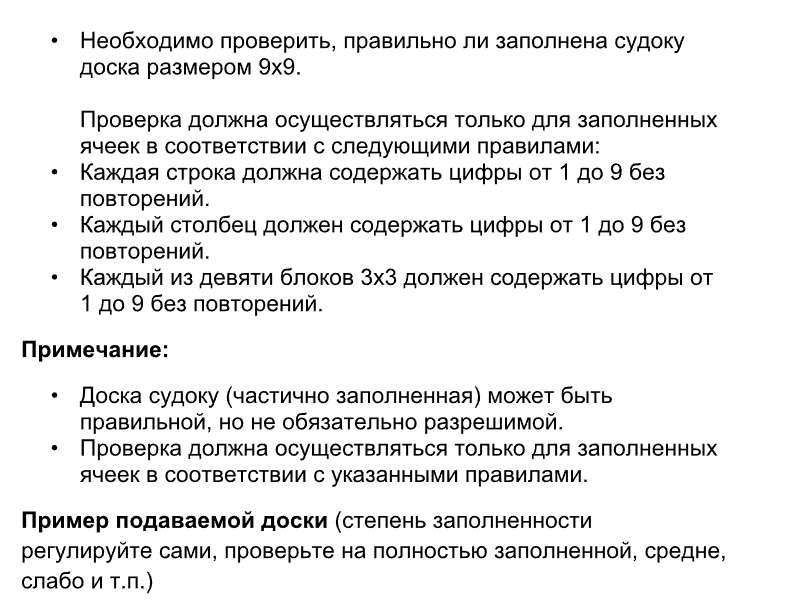
Маслов АН ИД23-1

Теория алгоритмов   
Домашнее задание  
Вариант 1

Задание 1



sudoku = [

[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,'','',7,9] ]

def valid\_check(sudoku):

    for row in range(9):

        for col in range(9):

            k = sudoku[row][col]

            if k == '': continue

            for j in range(9):

                if j != col and sudoku[row][j] == k:

                    return False

            for i in range(9):

                if i != row and sudoku[i][col] == k:

                    return False

            start\_row = (row // 3) \* 3

            start\_col = (col // 3) \* 3

            for i in range(3):

                for j in range(3):

                    r = start\_row + i

                    c = start\_col + j

                    if sudoku[r][c] == '': continue

                    if (r != row or c != col) and sudoku[r][c] == k:

                        return False

    return True

if (valid\_check(sudoku) == True):

    print("Cудоку заполнена правильно!")

else:

print("Судоку заполнена неправильно!")



Задание 2

Реализовать алгоритм нахождения наименьшего общего

кратного двух чисел и определить его временную и

пространственную сложность.

def gcd(a, b):

    while b != 0:

        a, b = b, a % b

    return a

def lcm(a, b):

    return abs(a \* b) // gcd(a, b)

print("НОК чисел 21 и 7: ", lcm(21, 7))



Временная сложность: O(log n) (один цикл, после которого происходит дополнительный расчёт)

Пространственная сложность: O(1) (нет рекурсий, или массивов)

Задание 3

Реализовать алгоритм поиска наибольшей возрастающей

последовательности в массиве и определить его временную и

пространственную сложность.

def lis(arr):

    max\_value = 0

    max\_arr = []

    for i in range(0, len(arr)):

        long\_arr = []

        long\_arr.append(arr[i])

        temp\_i = i

        for j in range(0, len(arr)):

            if arr[temp\_i] < arr[j] and j > temp\_i:

                temp\_i = j

                long\_arr.append(arr[j])

            else:

                continue

        if (len(long\_arr) > max\_value):

            max\_value = len(long\_arr)

            max\_arr = long\_arr

    return max\_value, max\_arr

print('Наибольшая возрастающая последовательности в массиве: ', lis([50, 3, 10, 7, 40, 80]))



Временная сложность: O(n2) (есть вложенный цикл)

Пространственная сложность: O(n) (есть массив значений)