

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ  
УНИВЕРСИТЕТ им. В. Г. ШУХОВА» (БГТУ им. В.Г. Шухова)



Кафедра программного обеспечения вычислительной техники и автоматизированных систем

### **Лабораторная работа №6**

по дисциплине: Исследования операций

тема: «Нахождение седловой точки в смешанных стратегиях для  
матричной игры с нулевой суммой»

Выполнил: ст. группы ПВ-223

Дмитриев А.А.

Проверил:

Вирченко Ю.П.

Белгород 2024 г.

**Цель работы:** Освоить метод нахождения седловой точки в смешанных стратегиях с помощью построения пары двойственных задач ЛП.

**Задания** для подготовки к работе:

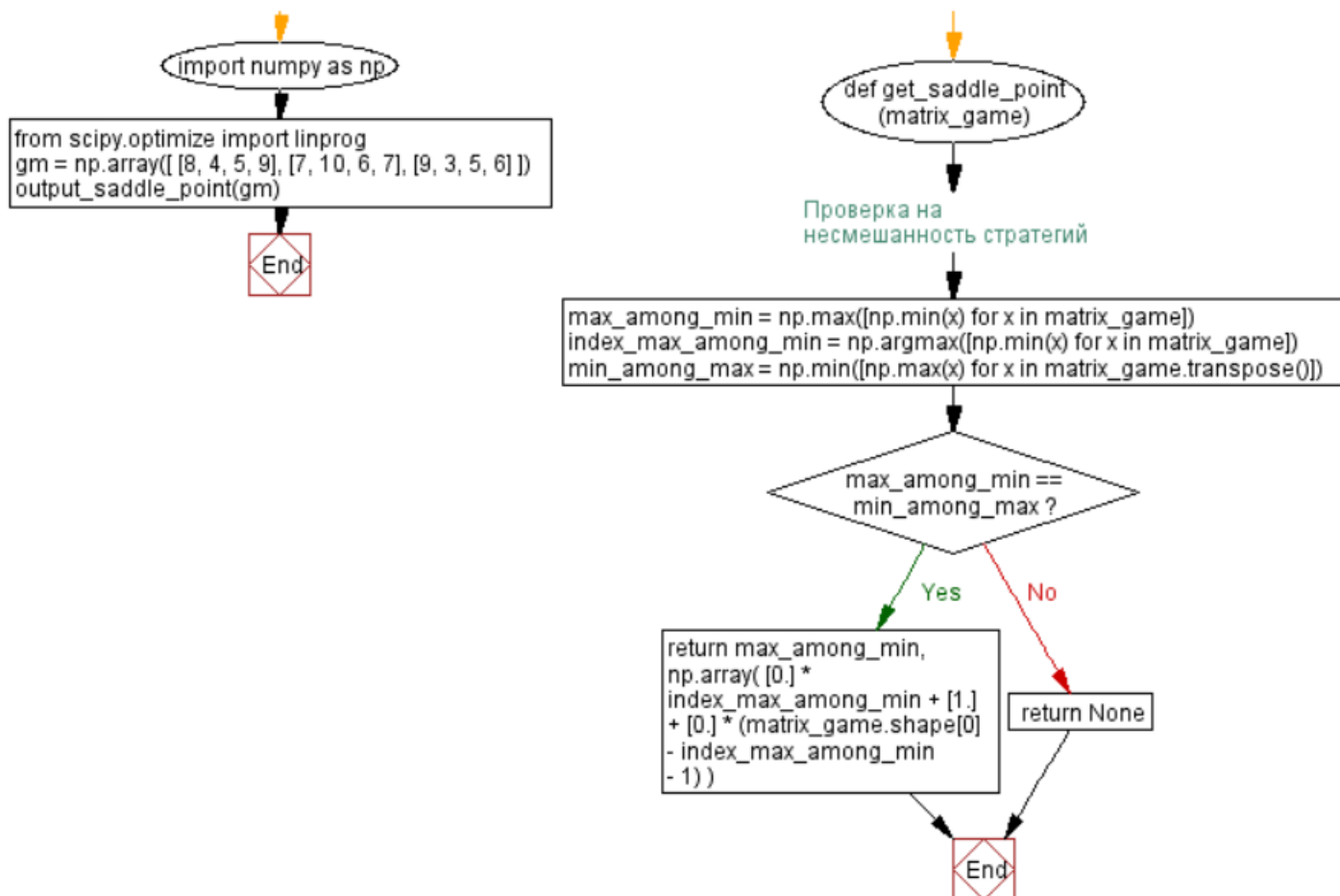
1. Изучить основные понятия теории матричных игр двух игроков с нулевой суммой, анализ игры в чистых стратегиях, понятие смешанной стратегии и седловой точки в смешанных стратегиях, а также метод нахождения седловой точки в смешанных стратегиях с помощью построения пары двойственных задач ЛП.
2. Составить и отладить программу для нахождения седловой точки игры с помощью решения пары симметрично двойственных задач ЛП.
3. Для подготовки тестовых данных решить вручную одну из задач.

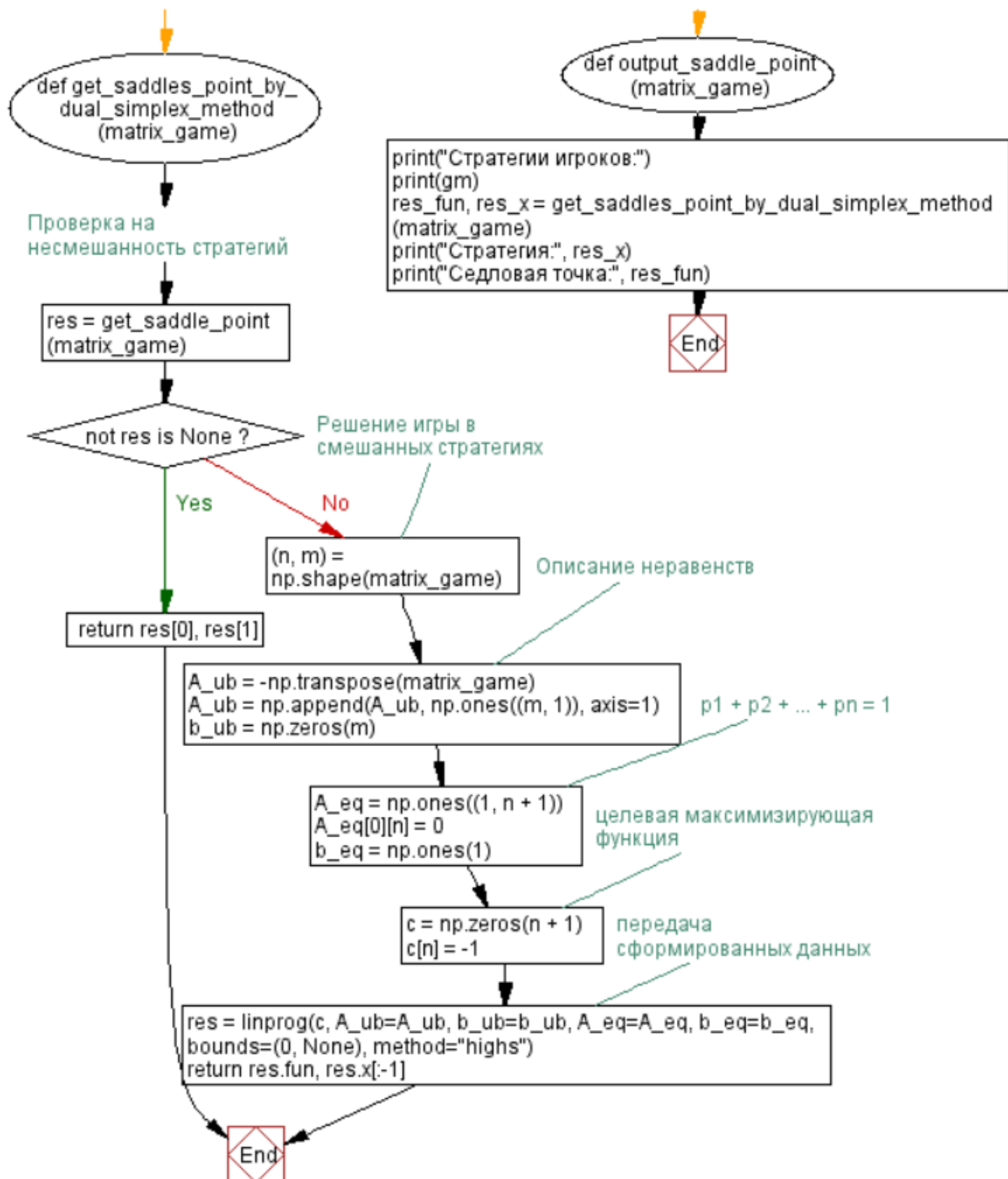
## Вариант 2:

$$\begin{pmatrix} 8 & 4 & 5 & 9 \\ 7 & 10 & 6 & 7 \\ 9 & 3 & 5 & 6 \end{pmatrix}$$

### Задание:

Блок схема:





## Листинг кода:

```
import numpy as np
from scipy.optimize import linprog

def get_saddle_point(matrix_game):
    # Проверка на несмешанность стратегий
    max_among_min = np.max([np.min(x) for x in matrix_game])
    index_max_among_min = np.argmax([np.min(x) for x in matrix_game])
    min_among_max = np.min([np.max(x) for x in matrix_game.transpose()])

    if max_among_min == min_among_max:
        return max_among_min, np.array(
            [0.] * index_max_among_min + [1.] + [0.] * (matrix_game.shape[0] -
            index_max_among_min - 1)
        )

    return None

def get_saddles_point_by_dual_simplex_method(matrix_game):
    # Проверка на несмешанность стратегий
    res = get_saddle_point(matrix_game)
    if not res is None:
        return res[0], res[1]

    # Решение игры в смешанных стратегиях
    (n, m) = np.shape(matrix_game)
    # Описание неравенств
    A_ub = -np.transpose(matrix_game)
    A_ub = np.append(A_ub, np.ones((m, 1)), axis=1)
    b_ub = np.zeros(m)
    # p1 + p2 + ... + pn = 1
    A_eq = np.ones((1, n + 1))
    A_eq[0][n] = 0
    b_eq = np.ones(1)
    # целевая максимизирующая функция
    c = np.zeros(n + 1)
    c[n] = -1
    # передача сформированных данных
    res = linprog(c, A_ub=A_ub, b_ub=b_ub, A_eq=A_eq, b_eq=b_eq, bounds=(0, None),
        method="highs")

    return res.fun, res.x[:-1]

def output_saddle_point(matrix_game):
    print("Стратегии игроков:")
    print(gm)
    res_fun, res_x = get_saddles_point_by_dual_simplex_method(matrix_game)
    print("Стратегия:", res_x)
    print("Седловая точка:", res_fun)

gm = np.array([
    [8, 4, 5, 9],
    [7, 10, 6, 7],
    [9, 3, 5, 6]
])

output_saddle_point(gm)
```

Тестовые данные:

Стратегии игроков:

[[ 8 4 5 9]

[ 7 10 6 7]

[ 9 3 5 6]]

Стратегия: [0. 1. 0.]

Седловая точка: 6.

**Решение вручную:**

$$\begin{pmatrix} 8 & 4 & 5 & 9 \\ 7 & 10 & 6 & 7 \\ 9 & 3 & 5 & 6 \end{pmatrix}$$

$$\max \min a_i = \max(4, 6, 3) = 6$$

$$\min \max b_j = \min(9, 10, 6, 9) = 6$$

Ответ: Седловая точка равна 6.

**Вывод:** В ходе лабораторной работы были изучены методы нахождения седловой точки в смешанных стратегиях с помощью построения пары двойственных задач ЛП.