

**Задание** состояло в численном решении антагонистической матричной игры, визуализации решения и сбора всего кода в пакет.

**Алгоритм** решения состоял в том, чтобы сначала проверить на седловую точку (fixed\_solution), а если её нет, то решать общим методом. Для общего метода мы решили не брать симплекс метод, а использовать “прямой” метод (mixed\_solution). “Прямой” метод заключается в том, что для каждого многогранника из пары двойственных задач находятся все крайние точки (так как матричные задачи образуют канонический многогранник). Далее из всех крайних точек находятся те, которые дают min/max сумму при умножении на единичный вектор. Таким образом решение матричной игры в общем случае сводится к нахождению крайних точек канонического многогранника размерности  $n \times n$  ( $n \times m$  – в прямоугольном случае). Алгоритм был придуман сам при поступлении на кафедру, как вступительная работа.

### Список участников и вклад:

Ловягин Андрей – “прямой” метод решения, ReadMe.pdf, тесты

Никита Денисов – решение через седловую точку, пакет, объединение частного и общего решения задачи

Иванков Михаил – математическая сторона программы, визуализация, вывод матрицы

### Использованные библиотеки:

NumPy – для работы с векторами и матрицами

itertools – для получения перестановок матриц и векторов, дабы облегчить жизнь и не писать это руками

re – для распарсивания входных данных

Fractions – для вывода в обыкновенных дробях

Matplotlib – для визуализации

Setuptools – для более удобного сбора пакета

### Инструкция по запуску (Только питон 3):

Тесты:

```
python3 -m "nose"
```

Использование из среды:

```
python
```

```

from game import nash_equilibrium
A = [[-1, -2, 3],[2, 4, 1]]
print(nash_equilibrium(A))

```

Вывод:

```

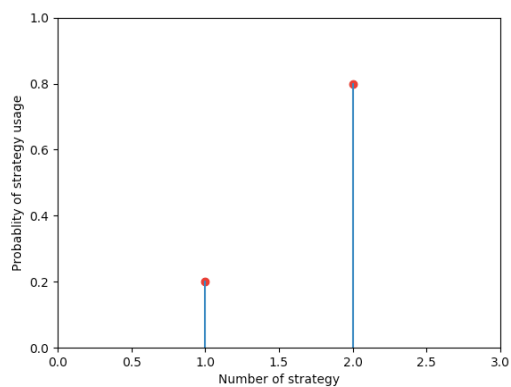
>>> print(nash_equilibrium(A))
No saddle point
| -1 | -2 | 3 |
|  2 |  4 | 1 |

Price of the game:  7/5

| p || 1/5 | 4/5 |
| q || 2/5 |  0 | 3/5 |

(array([1/5, 4/5]), array([2/5, 0, 3/5]), Fraction(7, 5))

```



x=2.96371 y=0.0113636

