

ІІТМО

Распределение весов в спортзале

Выполнили:

Поляков Александр Владимирович К3239

Синюков Лев Владимирович К3240

Лаптев Егор Игоревич К3239

Условия игры



1. Три слота в день:

- Тренировочный день состоит из 3 слотов, каждый из которых привязан к тренировке определенной группы мышц:

- T1 = Ноги

- T2 = Спина

- T3 = Грудь

2. У каждого спортсмена есть свои веса ценности для каждого слота

3. Выбор спортсменов:

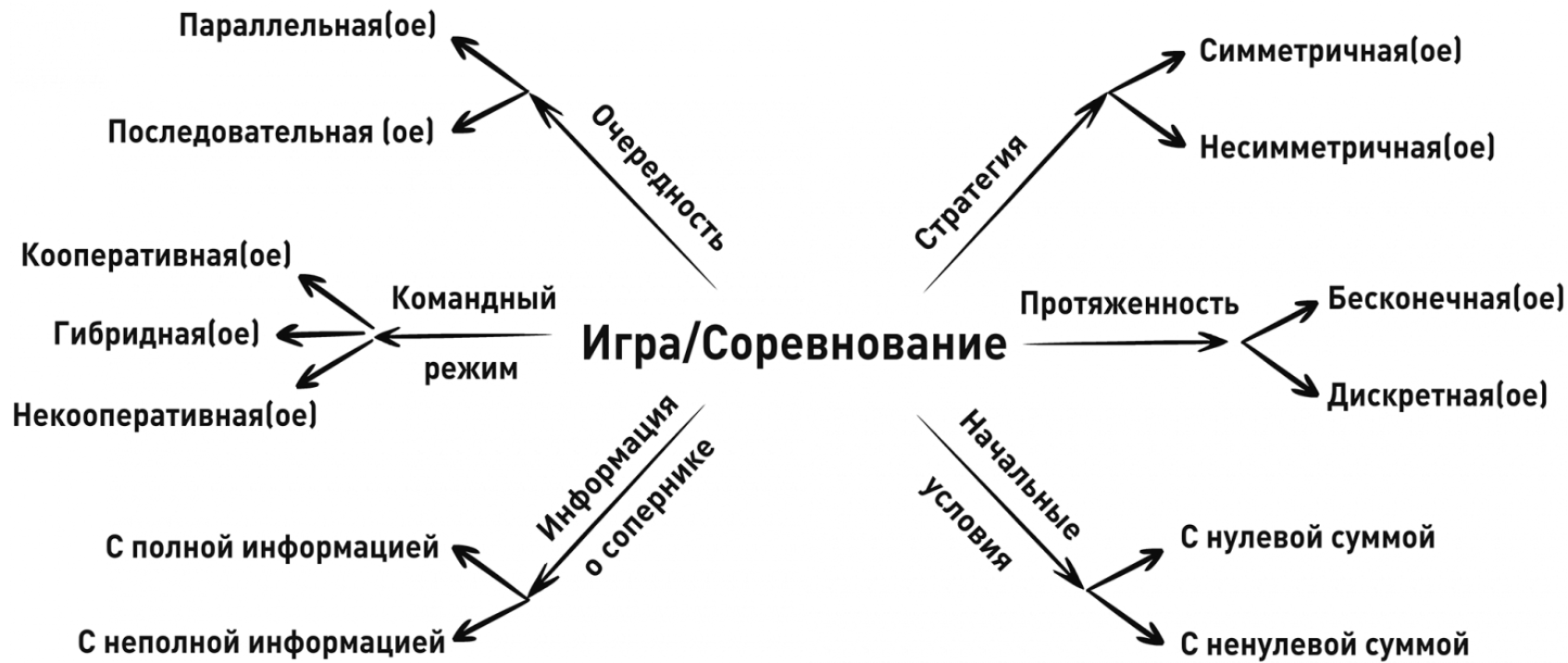
- Каждый игрок может выбрать максимум 2 слота в день;

- Выбранные слоты приносят ценность спортсмену.

4. Конфликты:

- Если оба спортсмена выбирают один и тот же слот, то каждый получает баллы с вероятностью отношения их ценностей за данный слот.

Типы игр



Типы игр



Типы игр



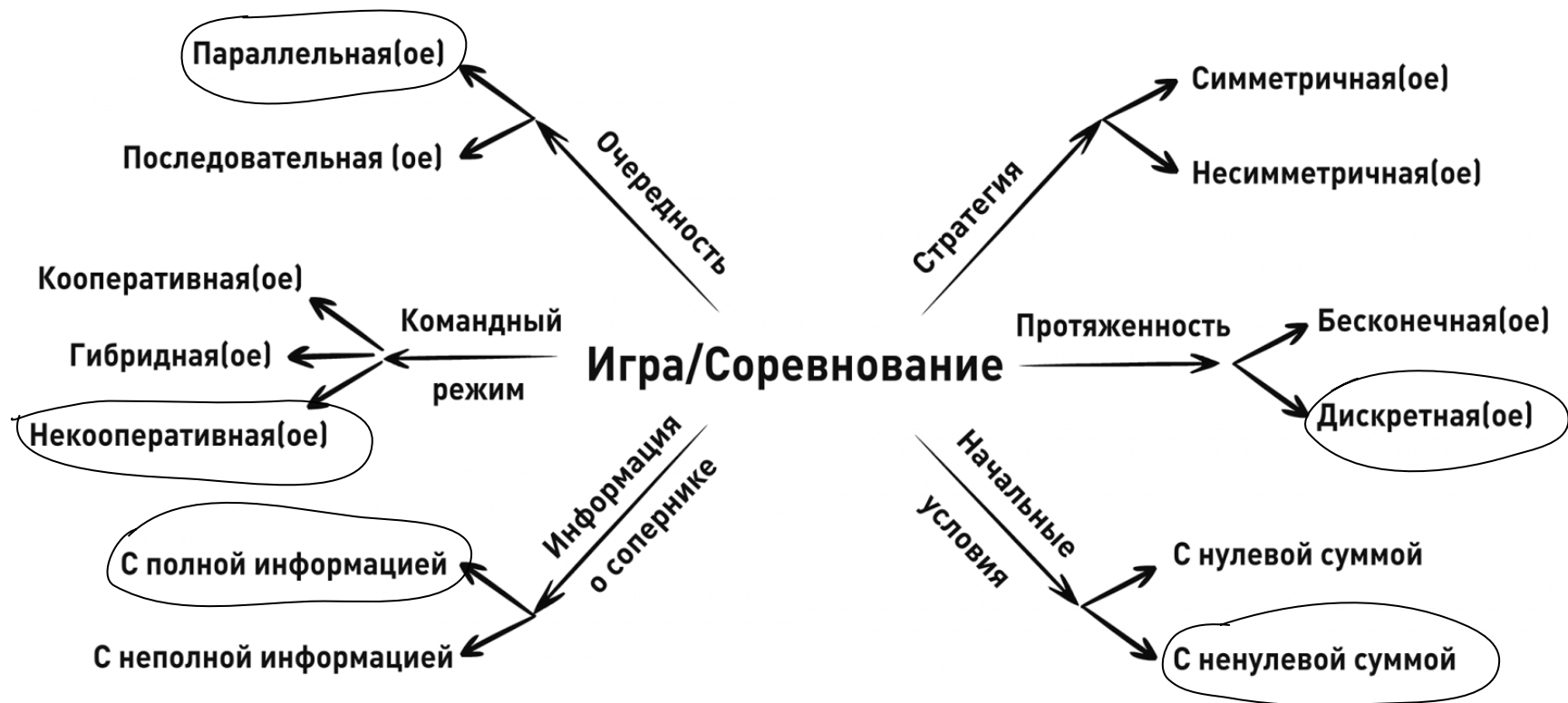
Типы игр



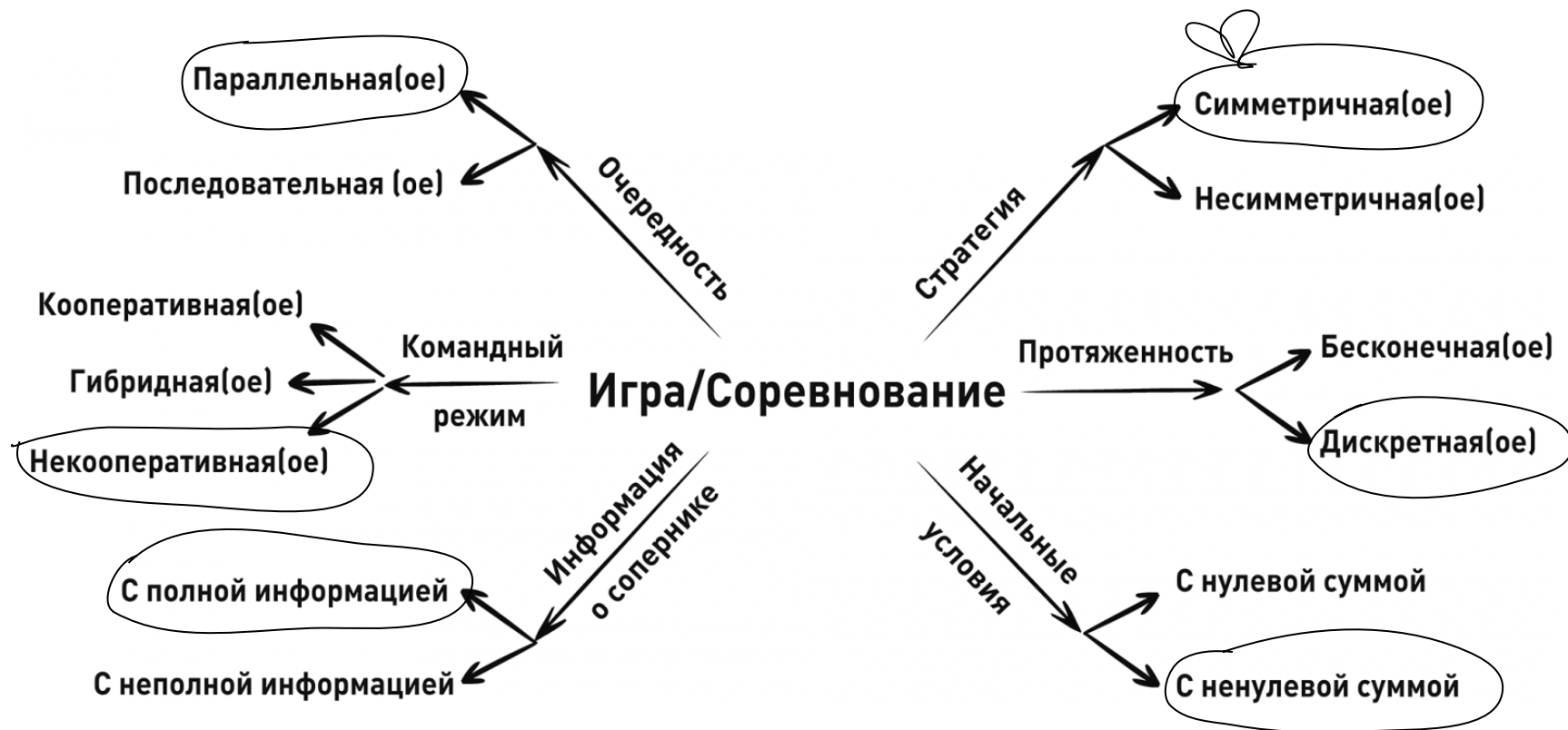
Типы игр

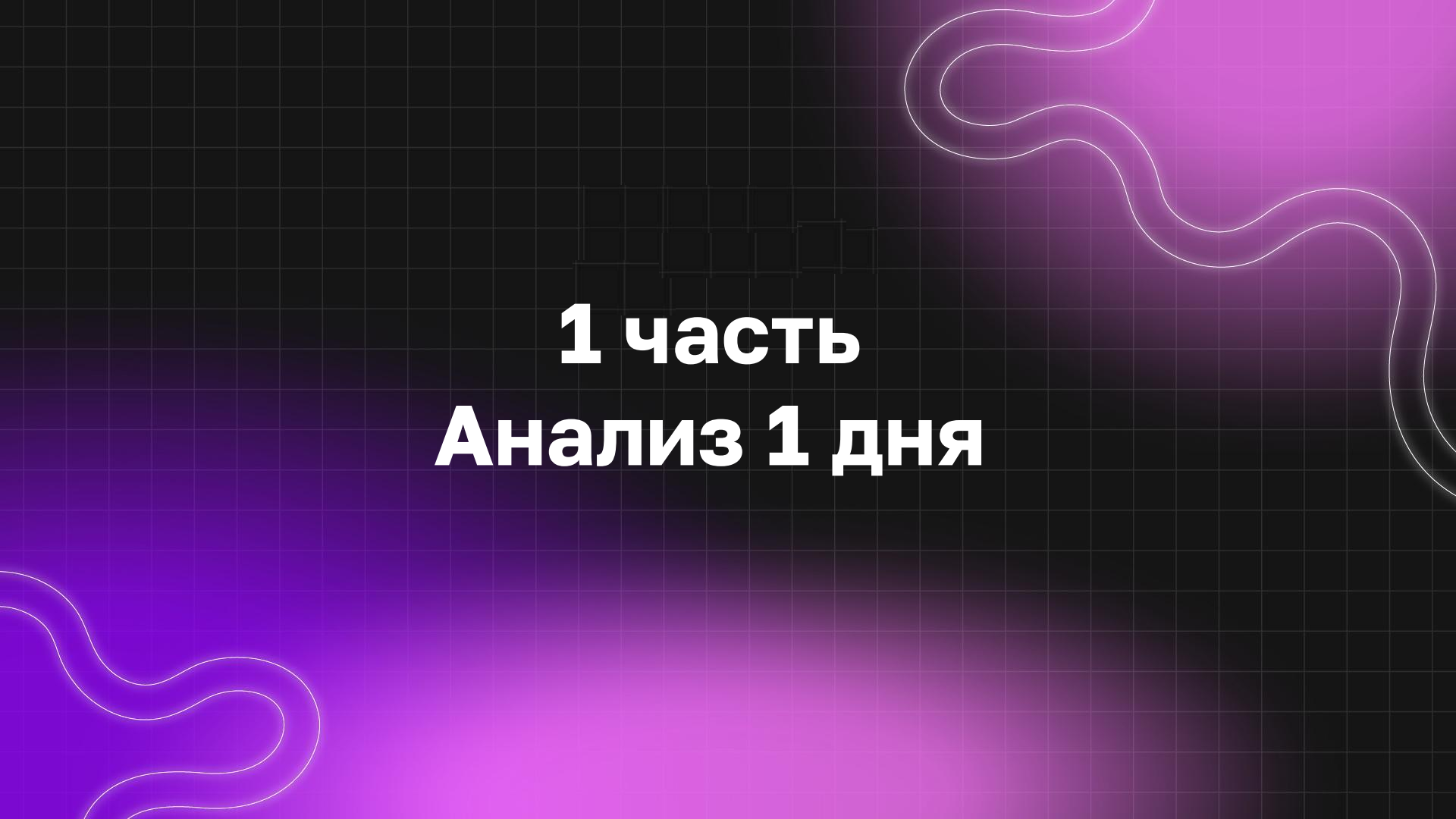


Типы игр



Типы игр





1 часть

Анализ 1 дня

Каждый игрок определяет приоритетность упражнений - для каждого упражнения определенная стоимость баллов.

Игрок 1:

$S_1 = 2;$

$S_2 = 2;$

$S_3 = 3$

Игрок 2:

$S_1 = 3;$

$S_2 = 1;$


$S_3 = 2$

Это позволит игрокам строить для себя выигрышную стратегию, отличную от оппонента;

В случае конфликта интересов (оба игрока выбрали одинаковое упражнение), баллы за упражнение достаются одному из игроков с вероятностью, равной отношению ценности данного слота для игрока к сумме ценностей за этот слот обоих игроков.

Пример: для игрока 1 - $S_1 = 2$, для игрока 2 - $S_1 = 3$. Тогда в случае конфликта с $p = \frac{2}{5}$ получит ценность (сможет потренироваться) игрок 1, а с $p = \frac{3}{5}$ игрок 2

Равновесие по Нэшу. Пример хода



	T1/T2	T1/T3	T2/T3
T1/T2	(2, 2)	(4, 3)	(4, 3)
T1/T3	(5, 2)	(2, 3)	(5, 2)
T2/T3	(5, 2)	(2, 5)	(3, 2)

Стратегия

Игрок 1: T1/T2, Игрок 2: T1/T3

Результат

→ **(4, 3)**

Оптимальность по Парето

```
def find_pareto_optimal(self):
    """ Identifies Pareto optimal outcomes from the payoff matrix. """
    outcomes = [(i, j), payoff] for i, row in enumerate(self.payoff_matrix) for j, payoff in enumerate(row)]
    pareto_optimal = []

    for ((i1, j1), (p1_payoff1, p2_payoff1)) in outcomes:
        if not any(
            p1_payoff2 >= p1_payoff1 and p2_payoff2 >= p2_payoff1 and
            (p1_payoff2 > p1_payoff1 or p2_payoff2 > p2_payoff1)
            for ((i2, j2), (p1_payoff2, p2_payoff2)) in outcomes
        ):
            pareto_optimal.append((p1_payoff1, p2_payoff1))

    return pareto_optimal
```

Алгоритм находит Парето-оптимальные исходы из матрицы выигрышей, перебирая все элементы. Сначала формируется список `outcomes`, содержащий координаты и пары выигрышей. Для каждого исхода $(p1_payoff1, p2_payoff1)$ проверяется, есть ли другой исход $(p1_payoff2, p2_payoff2)$, который доминирует его (т.е. выигрыши второго игрока не меньше, а одного из игроков – строго больше). Если доминирующих исходов нет, текущий добавляется в список `pareto_optimal`. В конце возвращается список всех недоминируемых исходов.

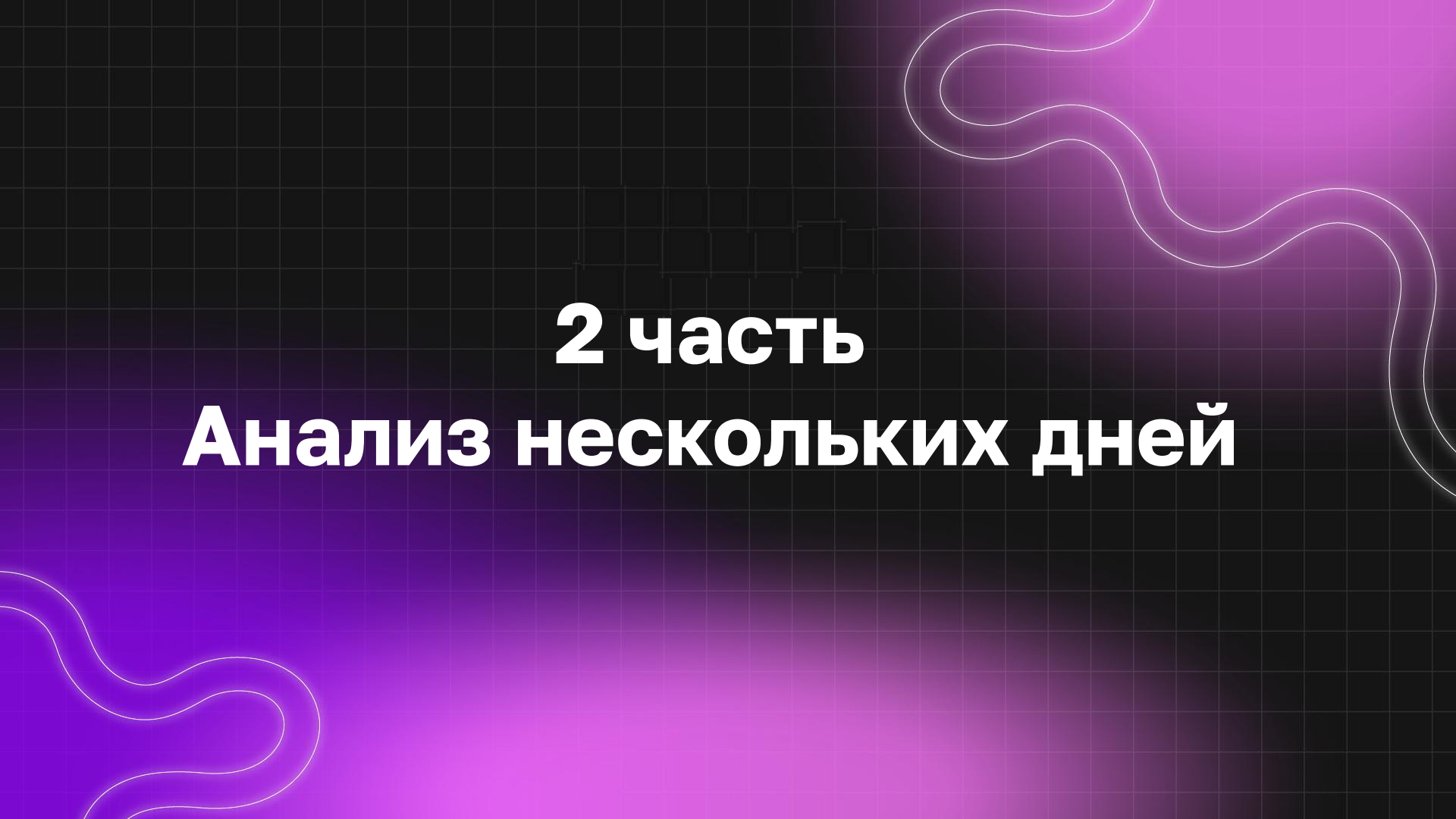
Оптимальность по Парето



	T1/T2	T1/T3	T2/T3
T1/T2	(2, 2)	(4, 3)	(4, 3)
T1/T3	(5, 2)	(2, 3)	(5, 2)
T2/T3	(5, 2)	(2, 5)	(3, 2)

Результат:

(4, 3), (4, 3), (5, 2), (5, 2), (5, 2), (2, 5)



2 часть

Анализ нескольких дней

Изменение условий

- Будем менять приоритеты игроков для каждого слота в зависимости от того, тренировали ли они эту группу мышц в последний день
 Если да - для данного слота ценность $\text{--} 1$
 Если нет - для данного слота ценность $\text{+} 1$

- Так как в случае конфликта, у нас используется вероятность при выборе, будет логично проверить работу алгоритма на одних и тех же входных данных (ценности игроков за каждый слот) на большом количестве дней несколько раз. Например, прогнать алгоритм 10 раз на 100 днях при одних и тех же данных.

Пример работы алгоритма для 10 дней

Steps 1-6:

```
(Step: 1) Payoff Matrix:
      T1/T2  T1/T3  T2/T3
T1/T2 (4, 0) (2, 5) (2, 5)
T1/T3 (3, 4) (3, 2) (2, 5)
T2/T3 (5, 2) (2, 5) (3, 2)

Nash Equilibrium: [NashResult(payoff=(3, 2), strategy_p1='T1/T3', strategy_p2='T1/T3')]
Pareto Optimal Outcomes: [(2, 5), (2, 5), (3, 4), (2, 5), (5, 2), (2, 5)]
(Step: 2) Payoff Matrix:
      T1/T2  T1/T3  T2/T3
T1/T2 (2, 2) (2, 5) (2, 5)
T1/T3 (3, 4) (5, 0) (2, 5)
T2/T3 (5, 2) (2, 5) (0, 5)

Nash Equilibrium: [NashResult(payoff=(2, 5), strategy_p1='T1/T2', strategy_p2='T2/T3'),
NashResult(payoff=(2, 5), strategy_p1='T1/T3', strategy_p2='T2/T3')]
Pareto Optimal Outcomes: [(2, 5), (2, 5), (3, 4), (2, 5), (5, 2), (2, 5)]
(Step: 3) Payoff Matrix:
      T1/T2  T1/T3  T2/T3
T1/T2 (2, 2) (2, 4) (1, 4)
T1/T3 (3, 4) (3, 2) (4, 2)
T2/T3 (3, 4) (5, 2) (0, 4)

Nash Equilibrium: [NashResult(payoff=(3, 4), strategy_p1='T1/T3', strategy_p2='T1/T2'),
NashResult(payoff=(3, 4), strategy_p1='T2/T3', strategy_p2='T1/T2')]
Pareto Optimal Outcomes: [(3, 4), (3, 4), (5, 2)]
```

```
(Step: 4) Payoff Matrix:
      T1/T2  T1/T3  T2/T3
T1/T2 (2, 1) (2, 3) (4, 2)
T1/T3 (4, 1) (2, 2) (2, 3)
T2/T3 (2, 2) (4, 1) (0, 3)

Nash Equilibrium: [NashResult(payoff=(2, 3), strategy_p1='T1/T2', strategy_p2='T1/T3')]
Pareto Optimal Outcomes: [(2, 3), (4, 2), (2, 3)]
(Step: 5) Payoff Matrix:
      T1/T2  T1/T3  T2/T3
T1/T2 (2, 0) (2, 1) (2, 1)
T1/T3 (2, 3) (2, 2) (3, 1)
T2/T3 (2, 3) (1, 3) (3, 0)

Nash Equilibrium: [NashResult(payoff=(2, 1), strategy_p1='T1/T2', strategy_p2='T1/T3'),
NashResult(payoff=(2, 3), strategy_p1='T1/T3', strategy_p2='T1/T2'), NashResult(payoff=(2, 3),
strategy_p1='T2/T3', strategy_p2='T1/T2')]
Pareto Optimal Outcomes: [(2, 3), (3, 1), (2, 3)]
(Step: 6) Payoff Matrix:
      T1/T2  T1/T3  T2/T3
T1/T2 (3, 0) (3, 1) (3, 1)
T1/T3 (1, 1) (2, 1) (3, 0)
T2/T3 (2, 1) (2, 1) (2, 0)

Nash Equilibrium: [NashResult(payoff=(3, 1), strategy_p1='T1/T2', strategy_p2='T1/T3'),
NashResult(payoff=(3, 1), strategy_p1='T1/T2', strategy_p2='T2/T3')]
Pareto Optimal Outcomes: [(3, 1), (3, 1)]
```

Пример работы алгоритма для 10 дней

Steps 7-10:

```
(Step: 7) Payoff Matrix:
      T1/T2  T1/T3  T2/T3
T1/T2 (0, 2) (1, 0) (1, 0)
T1/T3 (1, 2) (2, 0) (2, 0)
T2/T3 (1, 2) (1, 2) (1, 0)

Nash Equilibrium: [NashResult(payload=(1, 2), strategy_p1='T1/T3', strategy_p2='T1/T2'),
NashResult(payload=(1, 2), strategy_p1='T2/T3', strategy_p2='T1/T2')]
Pareto Optimal Outcomes: [(1, 2), (2, 0), (2, 0), (1, 2), (1, 2)]
(Step: 8) Payoff Matrix:
      T1/T2  T1/T3  T2/T3
T1/T2 (0, 1) (2, 0) (2, 0)
T1/T3 (2, 0) (0, 1) (2, 0)
T2/T3 (0, 1) (0, 1) (0, 0)

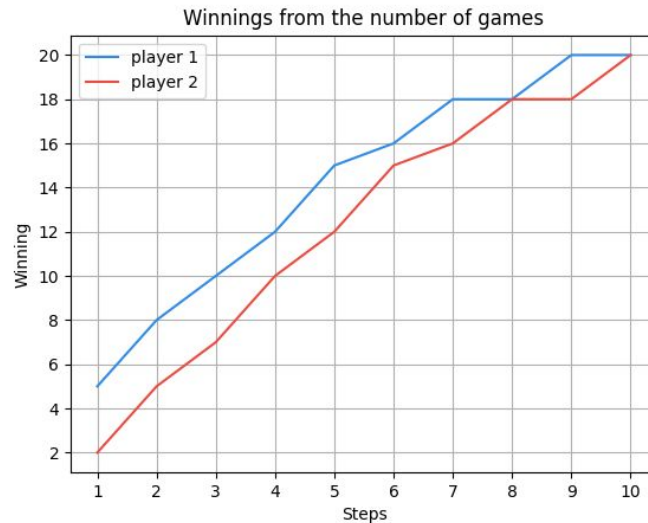
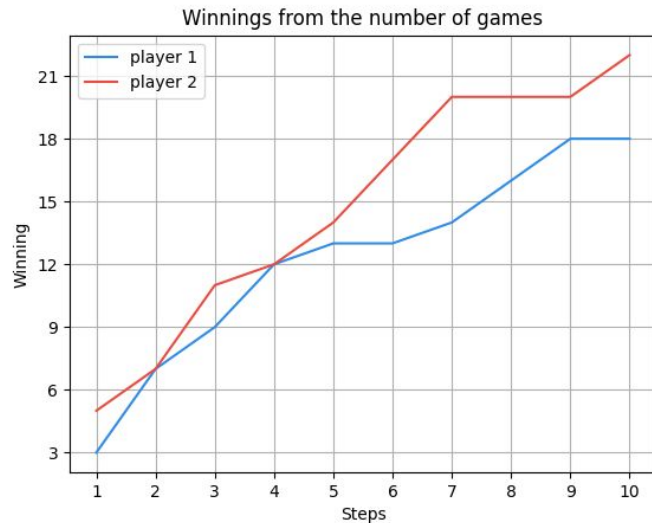
Nash Equilibrium: [NashResult(payload=(0, 1), strategy_p1='T1/T2', strategy_p2='T1/T2')]
Pareto Optimal Outcomes: [(0, 1), (2, 0), (2, 0), (2, 0), (0, 1), (2, 0), (0, 1), (0, 1)]
(Step: 9) Payoff Matrix:
      T1/T2  T1/T3  T2/T3
T1/T2 (0, 2) (1, 0) (1, 0)
T1/T3 (0, 2) (0, 2) (1, 0)
T2/T3 (0, 2) (0, 2) (0, 0)

Nash Equilibrium: [NashResult(payload=(0, 2), strategy_p1='T1/T2', strategy_p2='T1/T2'),
NashResult(payload=(0, 2), strategy_p1='T1/T3', strategy_p2='T1/T2'), NashResult(payload=(0, 2),
strategy_p1='T2/T3', strategy_p2='T1/T2')]
Pareto Optimal Outcomes: [(0, 2), (1, 0), (1, 0), (0, 2), (0, 2), (1, 0), (0, 2), (0, 2)]
(Step: 10) Payoff Matrix:
      T1/T2  T1/T3  T2/T3
T1/T2 (0, 1) (0, 1) (2, 0)
T1/T3 (2, 0) (2, 0) (2, 0)
T2/T3 (0, 1) (0, 1) (0, 0)

Nash Equilibrium: [NashResult(payload=(2, 0), strategy_p1='T1/T3', strategy_p2='T1/T2'),
NashResult(payload=(2, 0), strategy_p1='T1/T3', strategy_p2='T1/T3'), NashResult(payload=(2, 0),
strategy_p1='T1/T3', strategy_p2='T2/T3')]
Pareto Optimal Outcomes: [(0, 1), (0, 1), (2, 0), (2, 0), (2, 0), (2, 0), (0, 1), (0, 1)]
```

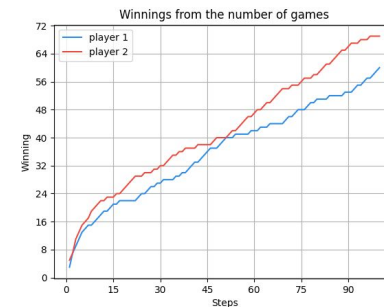
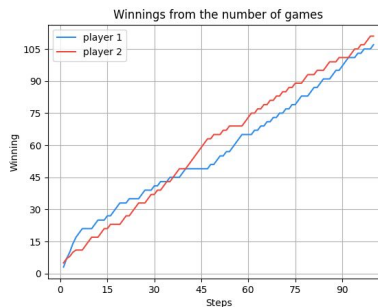
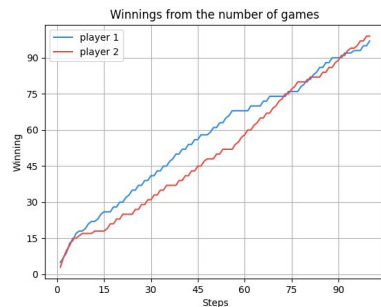
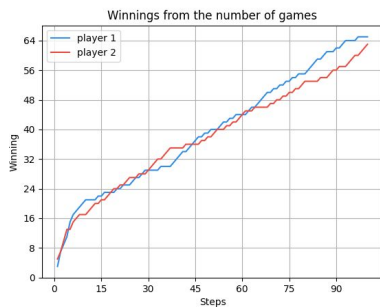
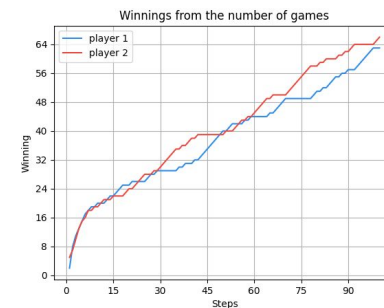
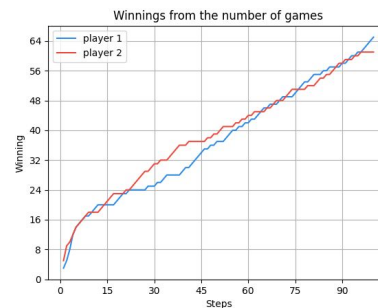
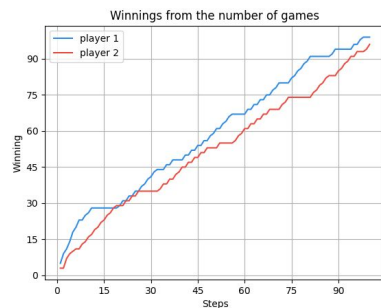
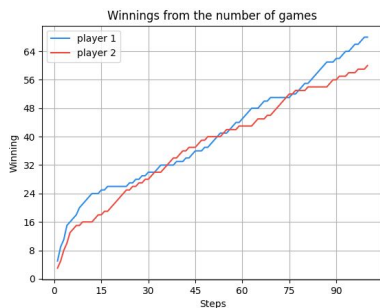
Пример работы алгоритма для 10 дней

График зависимости суммы выигрышных баллов от количества шагов



Пример запуска на данных большого объема

Запустим 10 раз симуляцию 100 дней на некоторых входных данных



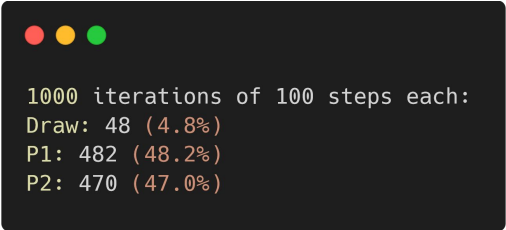
Пример запуска на данных большего объема



Как можно заметить по графикам, у нас нет конкретного доминирующего игрока, вероятность распределяется примерно одинаково на дистанции.

Также при тестировании программы, мы заметили, что при разных начальных условиях (разные наборы весов для слотов у каждого спортсмена), графики резко меняются лишь в первые несколько дней (разница в выигрышах заметна), а ближе к началу второго десятка дней зависимость становится постоянной.

По итогам тестирования у нас получились следующие процентные соотношения выигрышей игрока 1 и игрока 2 соответственно:



```
1000 iterations of 100 steps each:  
Draw: 48 (4.8%)  
P1: 482 (48.2%)  
P2: 470 (47.0%)
```

Таким образом, **начальные условия не сильно влияют на дальнейшее развитие игры.**

**Спасибо
за внимание!**

