ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ»

Факультет информационных технологий и программирования

Дисциплина:

«Прикладная математика»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2

Выполнил:

M33091 Ларин В. Д.

Проверила:

Москаленко М. А.

Санкт-Петербург

2021 г.

Задача 1.

1. Где строить? Две конкурирующие крупные торговые фирмы F_1 и F_2, планируют построить в одном из четырех небольших городов G_1, G_2, G_3, G_4, лежащих вдоль автомагистрали, по одному универсаму. Взаимное расположение городов, расстояние между ними и численность населения показаны на следующей схеме:

140 км	30 км	40 км	50 км	150 км
	G_1	G_2	G_3	G_4
Число покупателей	30 тыс	50 тыс	40 тыс	30 тыс

Доход, получаемый каждой фирмой, определяется численностью населения городов, а также степенью удаленности универсамов от места жительства потенциальных покупателей. Специально проведенное исследование показало, что доход универсамов будет распределяться между фирмами так, как это показано в следующей таблице:

Условие	F_1	F_2
Универсам фирмы F_1 расположен от города ближе универсама фирмы F_2	75%	25%
Универсамы обеих фирм расположены на одинаковом расстоянии от города	60%	40%
Универсам фирмы F_1 расположен от города дальше универсама фирмы F_2	45%	55%

Например, если универсам фирмы F_1 расположен от города G_1 ближе универсама фирмы F_2, то доход фирм от покупок, сделанных жителями данного города, распределится следующим образом: 75% получит F_1, остальное – F_2.

а) Представьте описанную ситуацию, как игру двух лиц;

б) В каких городах фирмам целесообразно построить свои универсамы?

Матрица прибыли F1:

	G1	G2	G3	G4
G1	90	76.5	91.5	91.5
G2	103.5	90	91.5	103.5
G3	88.5	88.5	90	103.5
G4	88.5	76.5	76.5	90

Матрица прибыли F2:

	G1	G2	G3	G4
G1	60	73.5	58.5	58.5
G2	46.5	60	58.5	46.5
G3	61.5	61.5	60	46.5
G4	61.5	73.5	73.5	60

```python
c = np.array([1, 1, 1, 1, 0, 0, 0, 0])  # * Коэффиценты функции
A = np.array([[90, 76.5, 91.5, 91.5, 1, 0, 0, 0],
              [103.5, 90, 91.5, 103.5, 0, 1, 0, 0],
              [88.5, 88.5, 90, 103.5, 0, 0, 1, 0],
              [88.5, 76.5, 76.5, 90, 0, 0, 0, 1]] )  # * Ограничения
b = np.array([1, 1, 1, 1])  # * Свободные коэффициенты
```

```
min_x = [0.         0.01111111 0.         0.         0.15       0.
 0.01666667 0.15       0.        ]
f(min_x) = 0.01
```

Компании F1 целесообразно выбирать город с наименьшим риском, то есть строку, в которой минимальное значение будет максимальным, - вторая относительно минимальных значений других строк.

Аналогично для компании F2, только в этот раз ищем по столбцам.

```python
c = np.array([1, 1, 1, 1, 0, 0, 0, 0])  # * Коэффиценты функции
A = np.array([[60, 46.5, 61.5, 61.5, 1, 0, 0, 0],
              [73.5, 60, 61.5, 73.5, 0, 1, 0, 0],
              [58.5, 58.5, 60, 73.5, 0, 0, 1, 0],
              [58.5, 46.5, 46.5, 60, 0, 0, 0, 1]] )  # * Ограничения
b = np.array([1, 1, 1, 1])  # * Свободные коэффициенты
```

```
min_x = [0.         0.01666667 0.         0.         0.225      0.
 0.025      0.225      0.        ]
f(min_x) = 0.02
```

В итоге получаем *седловую точку* (G2; G2)

Задача 2.

2. Двум погрузчикам разной мощности за 24 часа нужно погрузить на первой площадке 230 т, на второй - 68 т. Первый погрузчик на 1-ой площадке может погрузить 10 т в час, а на 2-ой - 12 т в час. Второй погрузчик на каждой площадке может погрузить по 13 т в час. Стоимость работ, связанных с погрузкой 1 т первым погрузчиком на первой площадке 8 руб., на второй - 7 руб., вторым погрузчиком на первой площадке - 12 руб., на второй - 13 руб. Нужно найти, какой объем работ должен выполнить каждый погрузчик на каждой площадке, чтобы стоимость всех работ по погрузке была минимальной.

x_1 - время, потраченное 1-м погрузчиком на 1-й площадке

x_2 - время, потраченное 1-м погрузчиком на 2-й площадке

x - время, потраченное 2-м погрузчиком на 1-й площадке

x - время, потраченное 2-м погрузчиком на 2-й площадке

$$f(x) = 80x_1 + 156x_2 + 84x_3 + 169x_4 \rightarrow min$$

$$\begin{cases} x_1 + x_2 + x_3 + x_4 + x_5 = 24 \\ 10x_1 + 13x_2 = 230 \\ 12x_3 + 13x_4 = 68 \end{cases}$$

$$x_i \geq 0$$

Таким образом вывод программы:

```
min_x = [ 2.77777778 15.55555556  5.66666667  0.          0.          0.        ]
f(min_x) = 3124.89
```

Таким образом минимально затраченное время 3124.89 стоимость работ

Задача 3.

3. При составлении суточного рациона кормления скота используют сено и силос. Рацион должен обладать определенной питательностью и содержать белка не менее 1 кг, кальция не менее 100 г и фосфора не менее 80 г. При этом количество питательного рациона должно быть не менее 60 кг. Содержание питательных компонентов в 1 кг сена и силоса приведено в следующей таблице. В ней указана также стоимость единицы того или иного корма. Требуется определить оптимальный суточный рацион кормления животных, обеспечивающий минимальную стоимость корма.

Название ингредиента	Норма (г)	Содержание ингредиента в 1 кг корма (г/кг)	
		Сено	Силос
Белок	1000	40	10
Кальций	100	1,25	2,5
Фосфор	80	2	1
Стоимость ед. корма (ден. ед.)		12	8

У нас даны все ограничения, составим функцию и систему неравенств по условию задачи: $f(x) = 12x_1 + 8x_2 \rightarrow min$,

$$\begin{cases} x_1 + x_2 \geq 60 \\ 2x_1 + x_2 \geq 80 \\ 1.25x_1 + 2.5x_2 \geq 100 \\ 40x_1 + 10x_2 \geq 1000 \end{cases}$$

$$x_i \geq 0$$

Приводим к каноническому виду:

```
c = np.array([12, 8, 0, 0, 0, 0])  # * Коэффиценты функции
A = np.array([[1, 1, -1, 0, 0, 0],
              [2, 1, 0, -1, 0, 0],
              [5, 4, 0, 0, -1, 0],
              [4, 1, 0, 0, 0, -1]])  # * Ограничения
b = np.array([60, 80, 40, 100])  # * Свободные коэффициенты
```

```
min_x = [ 20.  40.   0.   0. 220.  20.   0.]
f(min_x) = 560.00
```

У нас все готово для применения симплекс-метода, посчитаем результат:

Ответ: Исходя из полученного результата мы видим, что нам необходимо 2**0 кг сена и 40 кг силоса. И минимум функции равен 560.**

Задача 4.

4 задание

4. Пусть матрица проигрышей (в млн руб.) первого игрока имеет вид

$$\begin{pmatrix} 4 & 2 \\ 2 & 3 \end{pmatrix}$$

Решить матричную игру, перейдя к задаче линейного программирования. Найти оптимальную смешанную стратегию для первого игрока (использовать симплекс-метод).

Сведем задачу к ЛП:

$$\begin{cases} 4x_1 + 2x_2 \geq 1 \\ 2x_1 + 23x_2 \geq 1 \end{cases}$$

$F(x) = x_1 + x_2 \rightarrow min$

Используя симплекс-метод, решаем данную ЗЛП:

```
Result: [0.125 0.25  0.    0.   ]
Function value: 0.375
```

Нормализуя ответ, получаем: относительная частота первой стратегии - 0.333, относительная частота второй стратегии - 0.666
Ответ: [⅓, ⅔]

Задача 5.

5. Пусть матрица проигрышей (в млн руб.) первого игрока имеет вид

$$\begin{pmatrix} 8 & 4 & 6 \\ 4 & 8 & 5 \end{pmatrix}$$

Решить матричную игру, перейдя к задаче линейного программирования. Найти оптимальную смешанную стратегию для первого игрока (использовать симплекс-метод).

$$\begin{cases} 8x_1 + 4x_2 + 6x_3 \geq 1 \\ 4x_1 + 8x_2 + 5x_3 \geq 1 \end{cases}$$

$F(x) = x_1 + x_2 + x_3 \rightarrow min$

$x_j \geq 0$

```
Final dot: [0.083 0.083 0.    0.    0.083]
Target function: 0.167
```

$x1 = 1/12;\ x2 = 1/12$
Нормализуя, получаем ответ: [½, ½]

Задача 6.

6. Пусть матрица проигрышей первого игрока имеет вид

$$\begin{pmatrix} 7 & 2 & 5 & 1 \\ 2 & 2 & 3 & 4 \\ 5 & 3 & 4 & 4 \\ 3 & 2 & 1 & 6 \end{pmatrix}$$

Решить соответствующую матричную игру. Чему равно математическое ожидание проигрыша первого игрока, если и первый игрок, и второй игрок используют свои оптимальные стратегии?

Проверим матрицу проигрышей на наличие седловой точки, выделив получаем, что значение минимакса и максимина совпадают и равны 3. Следовательно, совпадает наибольшая и наименьшая цена игры -> мат.ожидание проигрыша первого игрока равно 3. Допустим, мы пропустили данную проверку, сведем матричную игру к задаче линейного программирования:

Тогда для выигрыша получим: $F(x) = x1 + x2 + x3 + x4 \rightarrow max$

$$7x_1 + 2x_2 + 5x_3 + 1x_4 \leq 1,$$
$$2x_1 + 2x_2 + 3x_3 + 4x_4 \leq 1,$$
$$5x_1 + 3x_2 + 4x_3 + 4x_4 \leq 1,$$
$$3x_1 + 2x_2 + 1x_3 + 6x_4 \leq 1,$$
$$x_j \geq 0$$

Посчитаем с помощью симплекс метода:

```
Final dot: [0.    0.333 0.    0.    0.333 0.333 0.    0.333]
Target function: 0.333
```

F(x) = ⅓
g = 1 : ⅓ = 3 - нижняя цена игры
p = (0, 1, 0, 0)

Теперь необходимо решить двойственную задачу линейного программирования**:**

$$F(x) = x_1 + x_2 + x_3 + x_4 \rightarrow min$$
$$7x_1 + 2x_2 + 5x_3 + 3x_4 \geq 1,$$
$$2x_1 + 2x_2 + 3x_3 + 2x_4 \geq 1,$$

$$5x_1 + 3x_2 + 4x_3 + 1x_4 \geq 1,$$
$$1x_1 + 4x_2 + 4x_3 + 6x_4 \geq 1,$$
$$x_j \geq 0$$

```
Final dot: [0.    0.    0.333 0.    0.667 0.    0.333 0.333]
Target function: 0.333
```

F(x) = ⅓

g = 1 : ⅓ = 3 - верхняя цена игры

q = (0, 0, 1, 0)

Посчитаем мат.ожидание**: E = p * A * q = 3**

Задача 7.

Пусть первый игрок придерживается следующей смешанной стратегии: (6/13, 3/13, 4/13), а второй (6/13, 4/13, 3/13). Вычислить математическое ожидание проигрыша первого игрока.

7. Платежная матрица в некоторой игре имеет вид

$$\begin{pmatrix} 1 & -1 & -1 \\ -1 & -1 & 3 \\ -1 & 2 & -1 \end{pmatrix}$$

Пусть первый игрок придерживается следующей смешанной стратегии: (6/13, 3/13, 4/13), а второй (6/13, 4/13, 3/13). Вычислить математическое ожидание проигрыша первого игрока.

Воспользуемся ф-лой: $M(p,q) = \sum_{i=1}^{m}\sum_{j=1}^{n} W(x,y)\,p_i q_j$, где pi - относительная частота стратегии 1-ого игрока, qi - относительная частота стратегии 2-ого игрока, A - платёжная матрица.

$$\begin{bmatrix} \frac{6}{13} & \frac{3}{13} & \frac{4}{13} \end{bmatrix} * \begin{bmatrix} 1 & -1 & -1 \\ -1 & -1 & 3 \\ -1 & 2 & -1 \end{bmatrix} * \begin{bmatrix} \frac{6}{13} & \frac{4}{13} & \frac{3}{13} \end{bmatrix} = \frac{1}{13} \approx 0.0769$$

Ответ: мат. ожидание проигрыша 1-ого игрока равно 1/13 или 0.0769

Задача 8.

8. Перейти от следующей задачи линейного программирования: $L(x) = x_1 + x_2 \to \min$

$$\begin{cases} 7x_1 + 2x_2 \geq 1, \\ x_1 + 11x_2 \geq 1, \\ x_1 \geq 0, \quad x_2 \geq 0 \end{cases}$$

к матричной игре. Решить Матричную игру любым известным способом.

Матрица
Пусть матрица выигрышей первого игрока имеет вид:

$$\begin{pmatrix} 7 & 2 \\ 1 & 11 \end{pmatrix}$$

$$\begin{cases} 7x_1 + x_2 \leq 1 \\ 2x_1 + 11x_2 \leq 1 \end{cases}$$
$$\boldsymbol{x_j \geq 0}$$

$F(x) = x_1 + x_2 \to max$

```
Final dot: [0.12 0.08 0.   0.  ]
Target function: 0.2
```

Получаем решение прямой задачи:
$x1 = 3/25;\ x2 = 2/25$
Находим линейную форму оптимальных планов как сумму найденных координат:
$F(x) = 1/5$

$$\begin{cases} 7y_1 + 2y_2 \geq 1 \\ y_1 + 11y_2 \geq 1 \end{cases}$$
$$\boldsymbol{y_j \geq 0}$$

$F(x) = y_1 + y_2 \to min$

```
Final dot: [0.133 0.067 0.    0.   ]
Target function: 0.2
```

Получаем решение двойственной задачи:

y1=2/15; y2=1/15

Находим линейную форму оптимальных планов как сумму найденных координат:

$G(x) = 1/5$

Находим цену игры:

$V = \frac{1}{F(x)} = \frac{1}{G(x)} = 5$

Находим оптимальную смешанную стратегию первого игрока:

$p = V * x_i = (0.6,\ 0.4)$

Находим оптимальную смешанную стратегию второго игрока:

$q = V * y_i = (0.666666667,\ 0.333333333)$

Задача 9.

9. Перейти от следующей задачи линейного программирования: $L(x) = x_1 + x_2 \to \max$

$$\begin{cases} 7x_1 + 2x_2 + 5x_3 + x_4 \le 1, \\ 2x_1 + 2x_2 + 3x_3 + 4x_4 \le 1, \\ 5x_1 + 3x_2 + 4x_3 + 4x_4 \le 1, \\ 3x_1 + 2x_2 + x_3 + 6x_4 \le 1, \\ x_1 \ge 0, \ldots, \quad x_2 \ge 0 \end{cases}$$

к матричной игре. Можно ли упростить матричную игру, используя понятие доминирования стратегий? Решить матричную игру любым известным вам способом.

Упростить можно: если *i-я строка* поэлементно <u>не меньше (≥) j-й строки</u>, то говорят, что **i-я строка доминирует над j-й строкой.** Поэтому игрок A не использует j-ю стратегию, так как его выигрыш при i-й стратегии не меньше, чем при j-й стратегии, вне зависимости от того, как играет игрок B. Аналогично, если *i-й столбец поэлементно не меньше (≥) j-го столбца*, то говорят, **что j-й столбец доминирует над i-м столбцом**. Поэтому игрок B не использует i-ю стратегию, так как его проигрыш (равный выигрышу игрока A) при j-й стратегии не больше (≤), чем при i-й стратегии, вне зависимости от того, как играет игрок A. Стратегии, над которыми доминируют другие стратегии, надо отбросить и приписать им нулевые вероятности. На цене игры это никак не скажется.

Проведем упрощение:

Можно удалить вторую строку, так как строка 3 доминирует над ней. Можно отбросить столбец 1, так как над ним доминирует столбец 3. Получаем новую систему неравенств:

$$2x_1 + 5x_2 + 1x_3 \le 1,$$
$$3x_1 + 4x_2 + 4x_3 \le 1,$$
$$2x_1 + 1x_2 + 6x_3 \le 1,$$
$$x_j \ge 0$$

Мы уменьшили размер матрицы с 4x4 до 3x3, теперь можно перейти к задаче линейного программирования: $F(x) = x_1 + x_2 + x_3 \to max$

$$2x_1 + 5x_2 + 1x_3 \le 1,$$
$$3x_1 + 4x_2 + 4x_3 \le 1,$$
$$2x_1 + 1x_2 + 6x_3 \le 1,$$
$$x_j \ge 0$$

```
Final dot: [0.333 0.    0.    0.333 0.    0.333]
Target function: 0.333
```

F(x) = ⅓

g = 1 : ⅓ = 3 - нижняя цена игры

p = (1, 0, 0)

Решим двойственную задачу линейного программирования:

$$F(x) = x_1 + x_2 + x_3 \to min$$

$$2x_1 + 3x_2 + 2x_3 \ge 1,$$
$$5x_1 + 4x_2 + 1x_3 \ge 1,$$
$$1x_1 + 4x_2 + 6x_3 \ge 1,$$
$$x_j \ge 0$$

```
Final dot: [0.    0.333 0.    0.    0.333 0.333]
Target function: 0.333
```

F(x) = ⅓

g = 1 : ⅓ = 3 - верхняя цена игры

q = (0, 1, 0)

Ответ: [0, 1, 0], [0, 1, 0]