Министерство образования Республики Беларусь  
Учреждение образования  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

КАФЕДРА ИНФОРМАТИКИ

Отчёт по лабораторной работе No2

По теме “Основная фаза симплекс метода”

Выполнила: студентка гр. 053503 Зырянова М.М.

Проверил: ассистент кафедры информатики Туровец Н. О.

Минск 2023

**СОДЕРЖАНИЕ**

[1 КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ 3](#_Toc134099392)

[2 АЛГОРИТМ ДВОЙСТВЕННОГО СИМПЛЕКС МЕТОДА 4](#_Toc134099393)

[3 ПРИМЕР РЕАЛИЗАЦИИ ОСНОВНОЙ ФАЗЫ СИМПЛЕКС МЕТОДА 6](#_Toc134099394)

[4 ДЕМОНСТРАЦИЯ РАБОТЫ ПРОГРАММЫ 11](#_Toc134099395)

[ПРИЛОЖЕНИЕ А 12](#_Toc134099396)

# **1 КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ**

Рассмотрим задачу линейного программирования в канонической форме:

,

где ∈ Rn — вектор переменных, A ∈ Rm×n — матрица, в которой m строк и n столбцов, b ∈ Rm. Будем предполагать, что система линейных ограничений Ax = b совместна и не содержит линейно зависимых ограничений, т.е. rank(A) = m.

Требуется определить ограничен ли сверху целевой функционал задачи на множестве допустимых планов и, в случае по- ложительного ответа, найти оптимальный план задачи. Это можно сделать с помощью основной фазы симплекс-метода. Цель настоящей лабораторной ра- боты — реализовать основную фазу симплекс-метода. Переходим к описанию метода.

Определение 1. Базисным допустимым планом задачи (1) называется упо- рядоченная пара (x,B), в которой первая компонента — это допустимый план x = (x1 , x2 , . . . , xn )⊺ задачи (1), а вторая — подмножество B множества индек- сов переменных {1, 2, . . . , n}, при этом:

а) (в множестве B ровно m индексов);

б) (базисная матрица AB — матрица, составленная из столбцов матрицы A с индексами из множества B, — имеет ненулевой определитель);

в) для каждого индекса i∈{1,2,...,n}\B выполняется xi =0.

Индексы переменных, принадлежащие множеству B, называются базисными, а остальные индексы — небазисными.

Известно, что базисных допустимых планов задачи (1) конечное число. Бо- лее того, если целевой функционал задачи ограничен сверху на множестве до- пустимых планов, то существует оптимальный план x, который вместе с соот- ветствующим множеством B базисных индексов является базисным допусти- мым планом.

Основная фаза симплекс-метода принимает на вход задачу (1) вместе с неко- торым базисным допустимым планом (x,B) и строит последовательность ба- зисных допустимых планов задачи (1), которая начинается с (x, B) и заканчи- вается (x∗,B∗), где x∗ — оптимальный план задачи (1), если он существует.

# **2 АЛГОРИТМ ДВОЙСТВЕННОГО СИМПЛЕКС МЕТОДА**

Вход: , , — параметры задачи (1) и (x, B) – начальный базисный допустимый план, причём элементы множества B упорядоченны.

Выход: оптимальный план x задачи (1) или сообщение о том, что целевой функционал задачи не ограничен сверху на множестве допустимых планов.

Шаг 1. Составим базисную матрицу и найдем для нее обратную мат- рицу .

Шаг 2. Сформируем вектор – вектор компонент вектора *с,* чьи индексы принадлежат множеству B.

Шаг 3. Находим вектор потенциалов

Шаг 4. Находим вектор оценок

Шаг 5. Проверим условие оптимальности текущего плана x, а именно, если ∆ ⩾ 0, то текущий x является оптимальным планом задачи (1) и метод завершает свою работу, возвращая в качестве ответа текущий x;

Шаг 6. Находим в векторе оценок ∆ первую отрицательную компоненту и ее индекс сохраним в переменной

Шаг 7. Вычислим вектор , – столбец матрицы А с индексом

Шаг 8. Находим вектор θ⊺ = (θ1,θ2,...,θm) ∈ Rm по следующему правилу

где — i-й по счету базисный индекс в упорядоченном наборе B.

Шаг 9. Вычислим



Шаг 10. Проверяем условие неограниченности целевого функционала: если то метод завершает свою работу с ответом «целевой функционал задачи (1) не ограничен сверху на множестве допустимых планов»;

Шаг 11. Находим первый индекс k, на котором достигается минимум в (2), и сохраним в переменной j∗ k-й базисный индекс из B;

Шаг 12. В упорядоченном множестве B заменим k-й индекс j∗ на индекс j0.

Шаг 13. Обновим компоненты плана x следующим образом: и для каждого i ∈ {1,2,...,m} такого, что

где — это i-й базисный индекс в B; xj∗ := 0. Переходим на Шаг 1.

# **3 ПРИМЕР РЕАЛИЗАЦИИ ОСНОВНОЙ ФАЗЫ СИМПЛЕКС МЕТОДА**

Рассмотрим задачу линейного программирования в канонической форме

x1 + x2 → max

Рассмотрим задачу:

x1 ⩾0

x2 ⩾0

x3 ⩾0

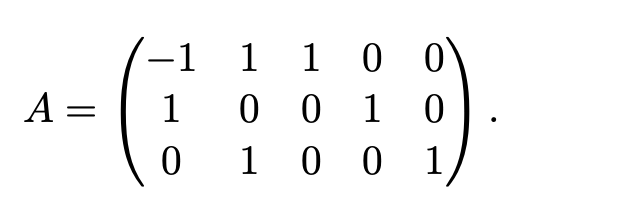
x4 ⩾0

x5 ⩾0

Вектор коэффициентов при переменных в целевом функционале:

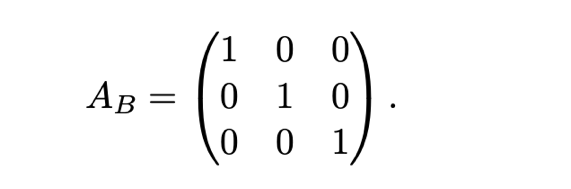
Вектор переменных:

Матрица коэффициентов при переменных в основных ограничениях:

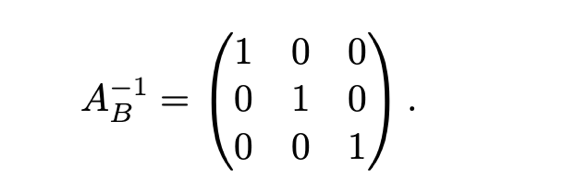


Пусть в качестве начального базисного допустимого плана выбрана пара (x⊺ =0 0 1 3 2, B = (j1 =3, j2 =4, j3 =5)). Базисными индексами являются индексы 3, 4 и 5, а небазисными — индексы 1 и 2.

Итерация 1. Составим базисную матрицу AB из столбцов матрицы A с базисными индексами (т.е. индексами из B). Матрица AB представляет собой матрицу, составленную из третьего, четвертого и пятого столбцов матрицы A.

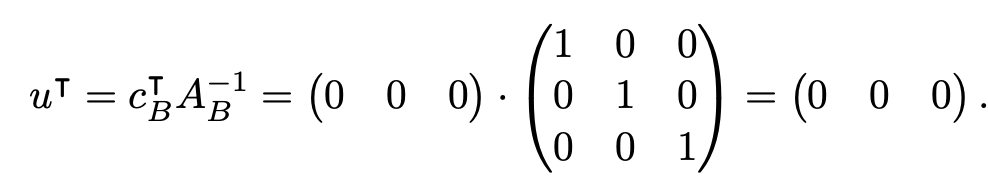


Находим матрицу, обратную к базисной .

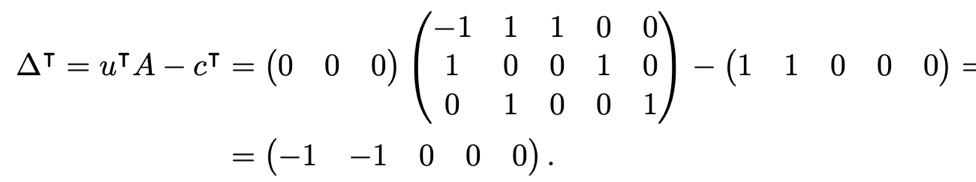


Формируем вектор , составленный из компонент вектора c с базисными ин- дексами .

Вычислим вектор потенциалов:

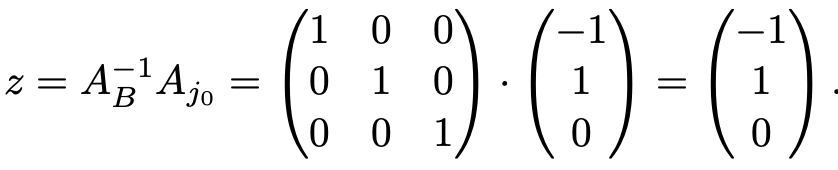


Находим вектор оценок



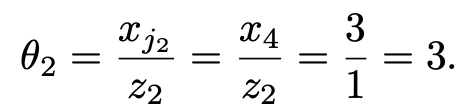
В векторе оценок ∆ выберем первую отрицательную компоненту и ее индекс сохраним в переменной : ∆1 = −1, = 1.

Вычислим вектор



Находим вектор θ⊺ = (θ1,θ2,θ3). Поскольку z1 = −1 < 0, то θ1 = ∞. Так как

z2 =1>0, то



Так как z3 = 0 ⩽ 0, то θ3 = ∞. Получаем θ⊺=(∞ 3 ∞.

Вычислим θ0 = min(∞, 3, ∞) = 3. Минимум достигается на θ2 . Поэтому k = 2, j∗ = jk = j2 = 4.

Обновим базисные индексы, т.е. в наборе B индекс j2 — индекс 4 — заменим на индекс j0:

B = (j1 =3, j2 = 4, j3 = 5) → B = (j1 = 3, j2 = 1, j3 = 5).

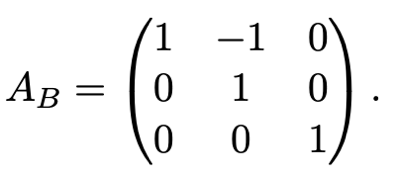
Обновим компоненты текущего плана x:

xj0 = x1= θ0 = 3  
xj1 = x3 := x3 − θ0 · z1 = 1− 3 · (−1) = 4

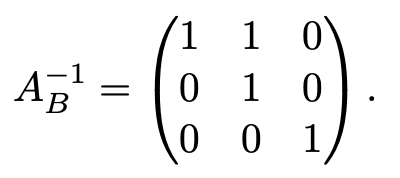
xj3 = x5 := x5 −θ0 · z3 = 2 – 3 · 0 = 2  
xj∗ = x4 = 0.

Получаем новый базисный допустимый план (x⊺ = (3 0 4 0 2), B = (j1 = 3, j2 = 1, j3 = 5)).

Итерация 2. Составим новую базисную матрицу

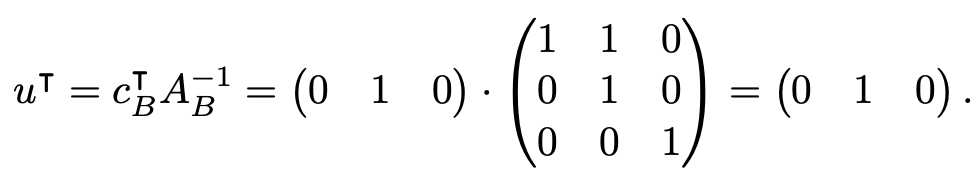


Заметим, что эта базисная матрица отличается от базисной матрицы на преды- дущей итерации только одним столбцом, а именно k-м (k = 2) столбцом. Для базисной матрицы на предыдущей итерации известна ее обратная матрица. Ис- пользуя метод из лабораторной работы No1, найдем матрицу, обратную к AB

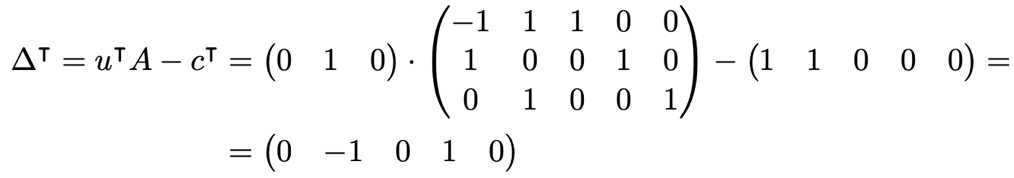


Формируем вектор из компонент вектора c с базисными индексами .

Находим вектор потенциалов

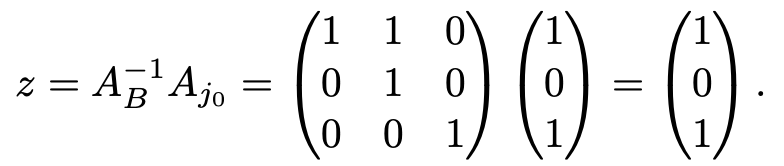


Вычислим вектор оценок

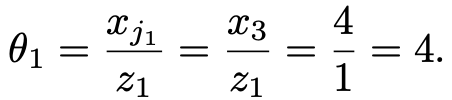


В векторе ∆ выберем первую отрицательную компоненту и ее индекс сохраним в переменной j0. Получаем ∆2 = −1 < 0, j0 = 2.

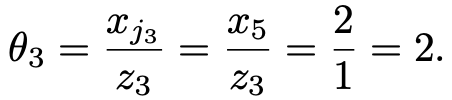
Вычислим вектор



Вычислим вектор θ⊺ = (θ1, θ2, θ3). Так как z1 = 1 > 0, то



Так как z2 = 0 ⩽ 0, θ2 = ∞. Так как z3 = 1 > 0, то



Найдем θ0 = min(θ1, θ2, θ3) = min(4, ∞, 2) = 2. Минимум достигается на θ3. Нас интересует третий базисный индекс, т.е. k = 3, j∗ = j3 = 5.

В наборе базисных индексов B заменим индекс j3, т.е. индекс 5, на индекс j0

B = (j1 = 3, j2 = 4, j3 = 5) → B = (j1 = 3, j2 = 1, j3 = 2).

Обновим компоненты текущего плана x:

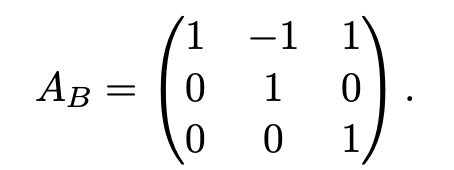
xj0 = x2 = θ0 = 2  
xj1 = x3 := x3 − θ0 · z1 = 4 – 2 · 1 = 2

xj2 = x1 := x1 − θ0 · z2 = 3 − 2 · 0 = 3

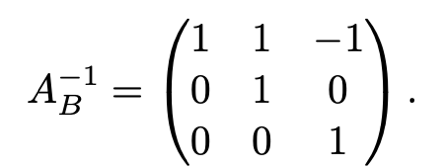
xj∗ = x5 = 0.

Получаем новый базисный допустимый план (x⊺ = (3 2 2 0 0), B = (j1 = 3, j2 = 1, j3 = 2)).

Итерация 3. Составим новую базисную матрицу



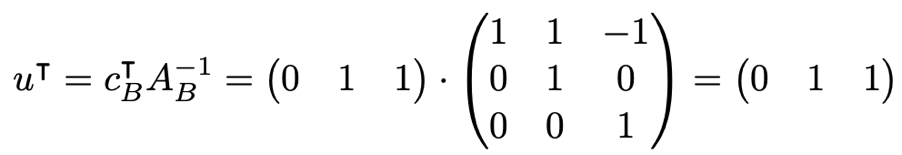
Новая базисная матрица отличается от базисной матрицы на предыдущей ите- рации только одним столбцом, а именно k-м (k = 3). Используя базисную мат- рицу из предыдущей итерации и ее обратную, методом обращения матрицы из лабораторной работы No1 найдем обратную матрицу для текущей базисной матрицы



Сформируем вектор cB из базисный компонент вектора c

.

Вычислим вектор потенциалов

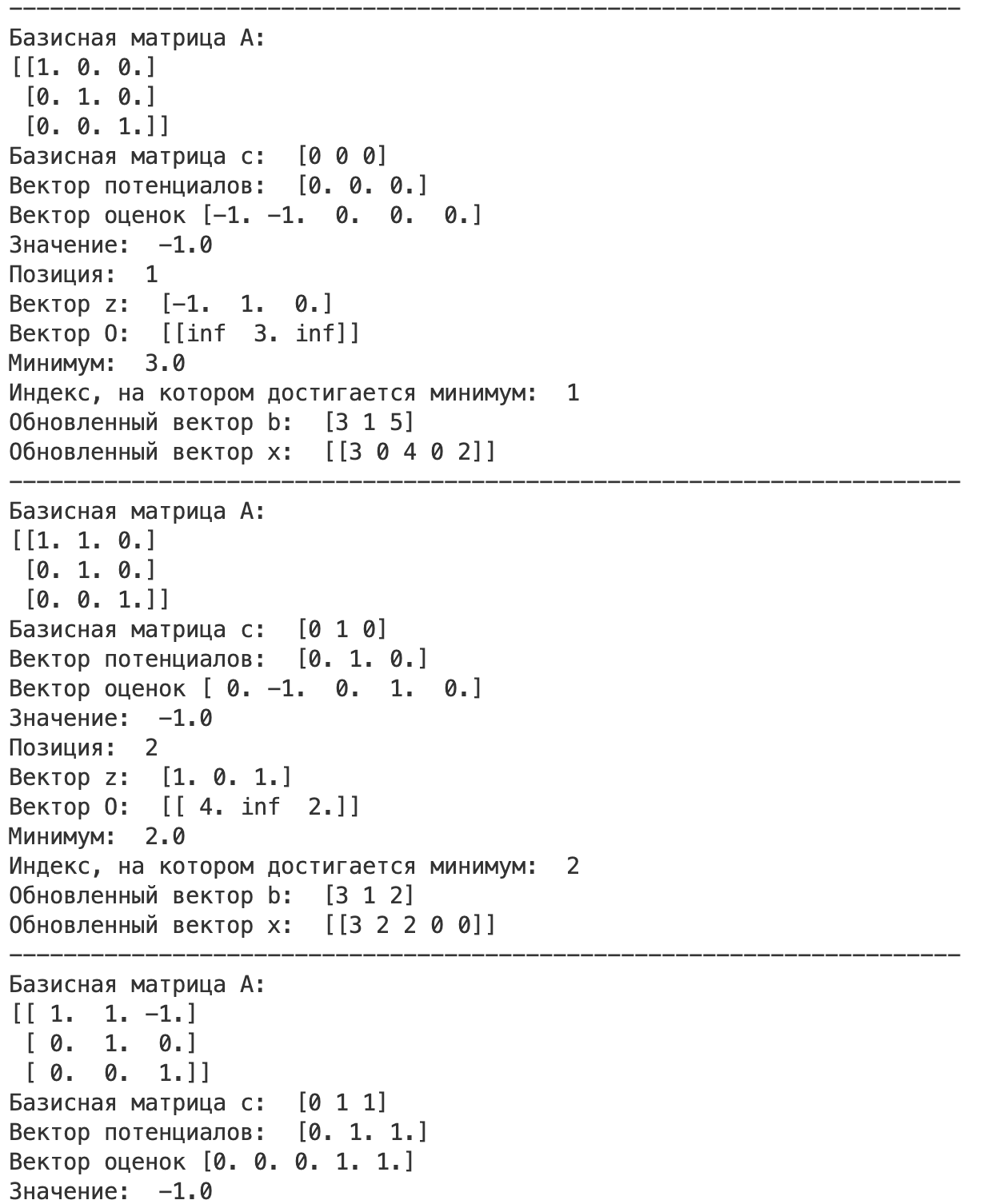


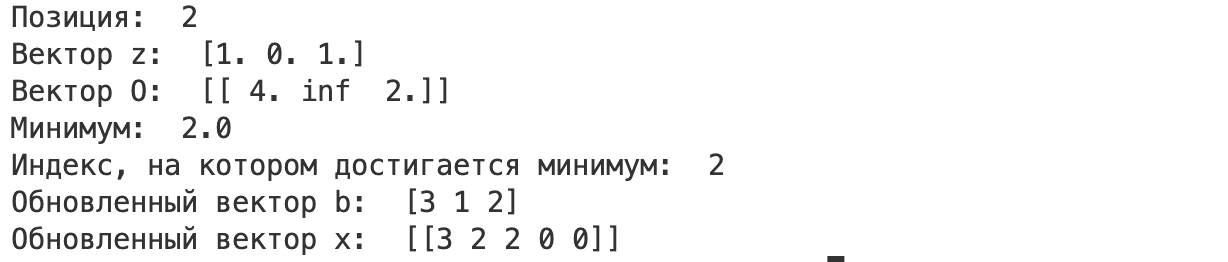
И вектор оценок: ∆⊺ = u⊺A − c⊺ = (0 0 0 1 1).

Так как ∆ ⩾ 0, то текущий x является оптимальным планом.

Ответ: x⊺ = (3 2 2 0 0).

# **4 ДЕМОНСТРАЦИЯ РАБОТЫ ПРОГРАММЫ**





# **ПРИЛОЖЕНИЕ А**

(Листинг программы)

import numpy as np

import math

def basis\_matrix(A: list, b: list) -> list:

A\_b = np.zeros((len(b), A.shape[0]))

i = 0

for k in b:

for j in range(len(A)):

A\_b[j][i] = A[j][k-1]

i += 1

return A\_b

def basis\_vector(c: list, b: list) -> list:

i = 0

c\_b = [0 for \_ in b]

for index in b:

c\_b[i] = c[index-1]

i += 1

return c\_b

def find\_negative(delta: list):

j = 0

for item in delta:

if item < 0:

return (item, j)

j += 1

def find\_tetta\_vector(z: list, b: list, x: list):

tetta = np.zeros(shape=(1,3))

minimum = 1000000000

index = 0

for i in range(len(z)):

if z[i] <= 0:

tetta[0][i] = math.inf

else:

tetta[0][i] = x[0][b[i]-1]/z[i]

if tetta[0][i] < minimum:

minimum = tetta[0][i]

index = i

return (tetta, minimum, index)

def update\_x(x: list, minimum: int, z: list, b: list, index: int, previous: int) -> list:

x[0][index-1] = minimum

x[0][previous-1] = 0

for i in range(len(b)):

if b[i] == index:

continue

x[0][b[i]-1] -= minimum\*z[i]

return x

def symplex(A: list, b: list, c: list, x: list):

while True:

basis\_A = basis\_matrix(A, b)

basis\_c = basis\_vector(c, b)

basis\_inverted\_A = np.linalg.inv(basis\_A)

basis\_transported\_c = np.transpose(basis\_c)

u = np.matmul(basis\_transported\_c,basis\_inverted\_A)

delta = np.matmul(u, A) - c

if np.all(delta >= 0):

print('-'\*70)

print('Базисная матрица А: ', basis\_inverted\_A, sep='\n')

print('Базисная матрица с: ', basis\_transported\_c)

print('Вектор потенциалов: ', u)

print('Вектор оценок', delta)

print('Значение: ', delta\_negative)

print('Позиция: ', delta\_negative\_position+1)

print('Вектор z: ', z)

print('Вектор O: ', tetta)

print('Минимум: ', minimum)

print('Индекс, на котором достигается минимум: ', index)

print('Обновленный вектор b: ', b)

print('Обновленный вектор x: ', x)

return(basis\_inverted\_A, x, b)

delta\_negative, delta\_negative\_position = find\_negative(delta)

z = np.array(np.matmul(basis\_inverted\_A, A[:, delta\_negative\_position]))

vect = find\_tetta\_vector(z, b, x)

tetta, minimum, index = vect

previous = b[index]

b[index] = delta\_negative\_position+1

x = update\_x(x, minimum, z, b, index, previous)

print('-'\*70)

print('Базисная матрица А: ', basis\_inverted\_A, sep='\n')

print('Базисная матрица с: ', basis\_transported\_c)

print('Вектор потенциалов: ', u)

print('Вектор оценок', delta)

print('Значение: ', delta\_negative)

print('Позиция: ', delta\_negative\_position+1)

print('Вектор z: ', z)

print('Вектор O: ', tetta)

print('Минимум: ', minimum)

print('Индекс, на котором достигается минимум: ', index)

print('Обновленный вектор b: ', b)

print('Обновленный вектор x: ', x)

if \_\_name\_\_ == "\_\_main\_\_":

A = np.array([

[-1, 1, 1, 0, 0],

[1, 0, 0, 1, 0],

[0, 1, 0, 0, 1]])

b = np.array([3, 4, 5])

c = np.array([1, 1, 0, 0, 0])

x = np.array([[0, 0, 1, 3, 2]])

symplex(A, b, c, x)