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

КАФЕДРА ИНФОРМАТИКИ

Отчёт по лабораторной работе No1

По теме “Обращение матрицы с измененным столбцом”

Выполнила: студентка гр. 053503 Зырянова М.М.

Проверил: ассистент кафедры информатики Туровец Н. О.

Минск 2023

**СОДЕРЖАНИЕ**

[1 КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ 3](#_Toc134054295)

[2 ПРИМЕР РЕАЛИЗАЦИИ АЛГОРИТМА ОБРАЩЕНИЯ МАТРИЦЫ 4](#_Toc134054296)

[3 ДЕМОНСТРАЦИЯ РАБОТЫ ПРОГРАММЫ 6](#_Toc134054297)

[ПРИЛОЖЕНИЕ А 7](#_Toc134054298)

# **1 КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ**

Пусть имеется обратимая квадратная матрица A порядка n вместе со своей обратной матрицей A−1. Пусть также задан вектор-столбец x высоты n. В матрице A заменяется i-й столбец на столбец x. В результате получается мат- рица A. Матрица A отличается от матрицы A только одним столбцом. Задача состоит в том, чтобы выяснить является ли матрица A обратимой; если матрица A обратима, то найти матрицу (A)−1, обратную к ней.

Для решения этой задачи можно использовать стандартные методы обращения матрицы, игнорируя тот факт, что обращаемая матрица A не сильно отличается от матрицы A, для которой обратная матрица уже известна. Существует более эффективный метод решить задачу, который существенно использует наличие дополнительных данных. Метод состоит в следующем:

Шаг 1. Находим . Если , то матрица A необратима и метод завершает свою работу; в противном случае матрица A обратима и мы переходим на следующий шаг.

Шаг 2. Формируем вектор , который получается из вектора l заменой i-го элемента на −1.

Шаг 3. Находим

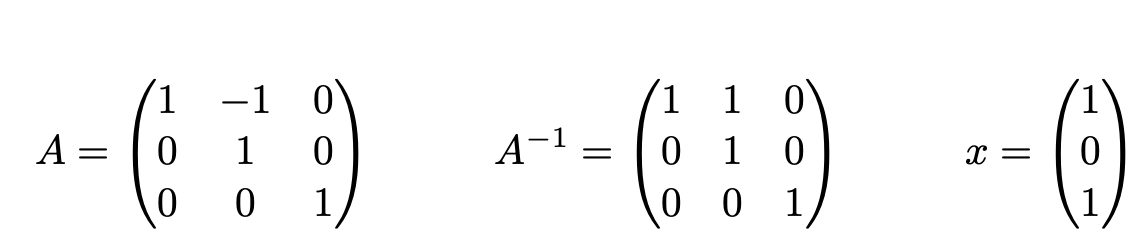
Шаг 4. Формируем матрицу Q, которая получается из единичной матрицы порядка n заменой i-го столбца на столбец .

Шаг 5. Находим

Шаги 1 – 4 выполняются за время O(n2). На шаге 5 умножаются две квадратные матрицы порядка n. Умножение двух таких матриц «по определению» занимает время O(n3). Матрица Q разреженная и имеет простую структуру. Это позволяет реализовать шаг 5 таким образом, чтобы его время работы было O(n2). Каждая строка матрицы Q содержит не более двух ненулевых элемен- тов. В j-й строке матрицы Q один из ненулевых элементов располагается на j-й позиции, а другой элемент, если он есть, — на i-ой позиции. Таким образом, для того, чтобы умножить j-ую строку матрицы Q на k-ый столбец матрицы A−1 достаточно умножить i-ый и j-ый элементы j-ой строки соответственно на i-ый и j-ый элементы k-го столбца, после чего получившиеся произведения сложить. В результате получим элемент матрицы , стоящий на пересечении j-ой строки и k-го столбца. Для нахождения одного элемента матрицы мы совершаем константное число арифметических операций (а именно, два умножения и одно сложение). Для вычисления всех n2 элементов матрицы нам понадобится совершить O(n2) арифметических операций.

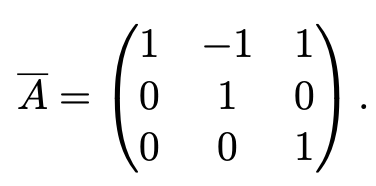
# **2 ПРИМЕР РЕАЛИЗАЦИИ АЛГОРИТМА ОБРАЩЕНИЯ МАТРИЦЫ**

Пусть

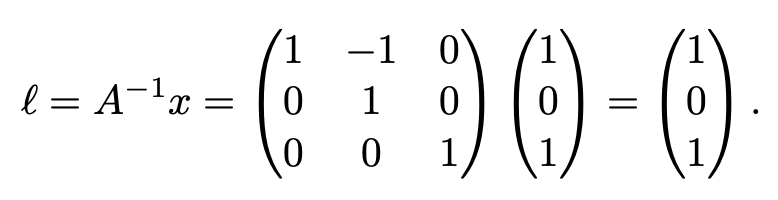


В матрице A заменим третий столбец (i = 3) на столбец x. В результате

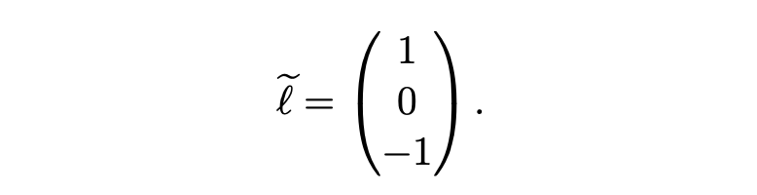
получим матрицу



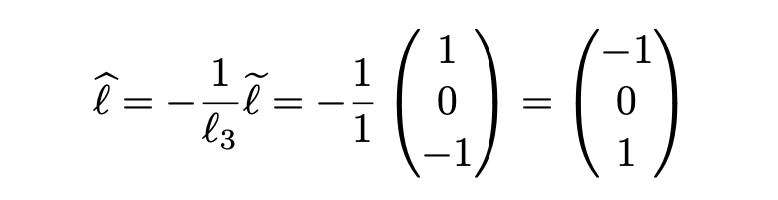
Требуется определить обратима матрица A и в случае положительного от- вета найти для нее обратную матрицу (A)−1. Находим вектор:



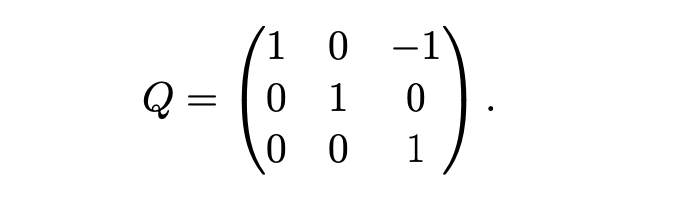
Поскольку то матрица A обратима. В копии вектора l заменим третий элемент на −1.



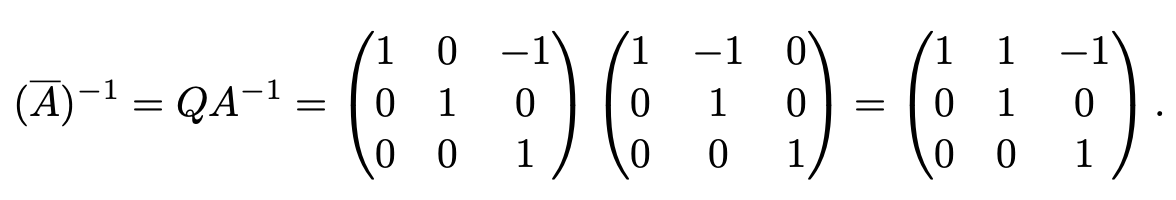
Находим:



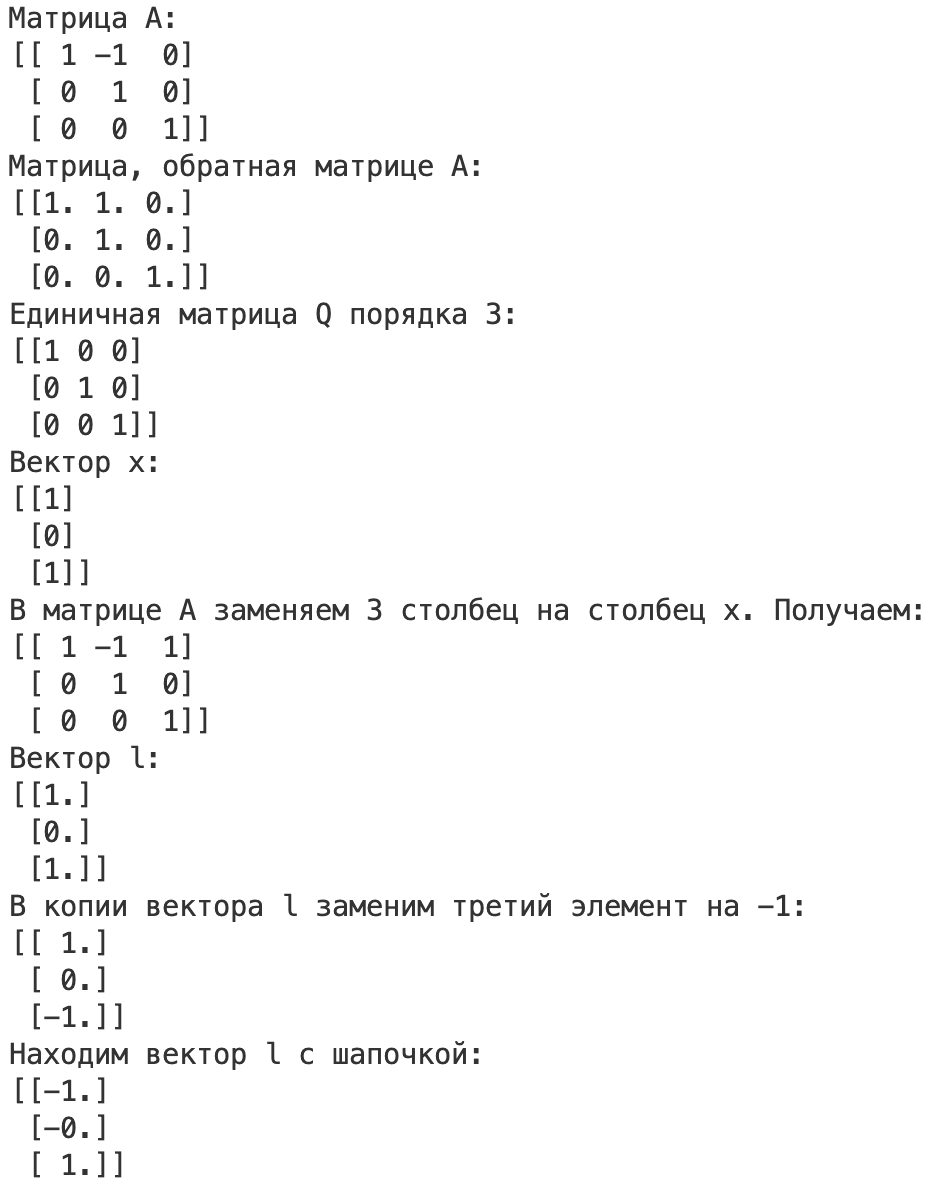
Заменим в единичной матрице порядка три третий столбец на столбец .

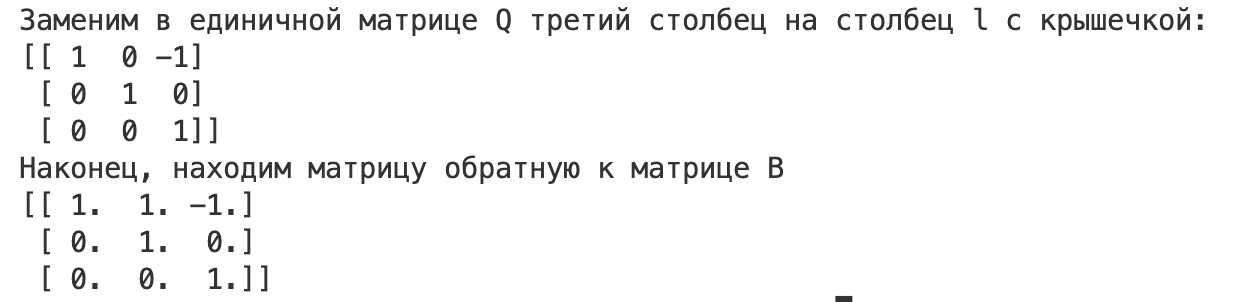


Наконец, находим матрицу обратную к матрице



# **3 ДЕМОНСТРАЦИЯ РАБОТЫ ПРОГРАММЫ**





# **ПРИЛОЖЕНИЕ А**

(Листинг программы)

import numpy as np

def change\_rows(m: np.matrix, x: np.matrix, column: int) -> np.matrix:

column -= 1

for c in range(len(x)):

m[c, column] = x[c, 0]

return m

if \_\_name\_\_ == '\_\_main\_\_':

A = np.matrix([

[1, -1, 0],

[0, 1, 0],

[0, 0, 1]])

print('Матрица А: ', A, sep='\n')

A\_opposit = np.linalg.inv(A)

print('Матрица, обратная матрице А: ', A\_opposit, sep='\n')

Q = np.matrix([

[1, 0, 0],

[0, 1, 0],

[0, 0, 1]

])

print('Единичная матрица Q порядка 3: ', Q, sep='\n')

x = np.matrix([[1], [0], [1]])

print('Вектор х: ', x, sep='\n')

i = 3

B = change\_rows(A, x[:, 0], i)

print(f'В матрице А заменяем {i} столбец на столбец х. Получаем: ', B, sep='\n')

l = A\_opposit\*x

print('Вектор l: ', l, sep='\n')

l\_new = l.copy()

l\_new[i-1] = -1

print('В копии вектора l заменим третий элемент на -1: ', l\_new, sep='\n')

l\_vect = -l\_new/l[i-1]

print('Находим вектор l с шапочкой:', l\_vect, sep='\n')

Q = change\_rows(Q, l\_vect, i)

print('Заменим в единичной матрице Q третий столбец на столбец l с крышечкой: ', Q, sep='\n')

reђult = Q.dot(A\_opposit)

print('Наконец, находим матрицу обратную к матрице B', result, sep='\n')