Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №7**

По дисциплине: «Языки программирования»

Тема: «Изучение NumPy. Сравнение производительности с классическими библиотеками Python»

Выполнила:

Студентка 2 курса

Группы ПО-7

Малинин Е.А

Проверил:

Бойко Д.О.

Брест, 2021

**Цель работы:**изучить NumPy. Сравнить производительность склассическими библиотеками Python

**Ход работы:**

Общие правила выполнения задания:

1. Для написания кода использовать библиотеки классического Python, NumPy и SciPy.

2. Код демонстрируется в Colaboratory

3. По каждому заданию должно быть предоставлено не менее 3-х вариантов решения, среди которых:

a) чистый NumPy (максимально оптимизованный, векторизованный)

b) любой не векторизованный вариант

c) любой другой вариант, желательно конкурентноспособный

4. Все варианты решения должны быть протестированы на скорость выполнения при помощи %timeit

5. Полученные результаты отразить в отчете и сделать выводы о производительности и комфорте использования NumPy в различных задачах.

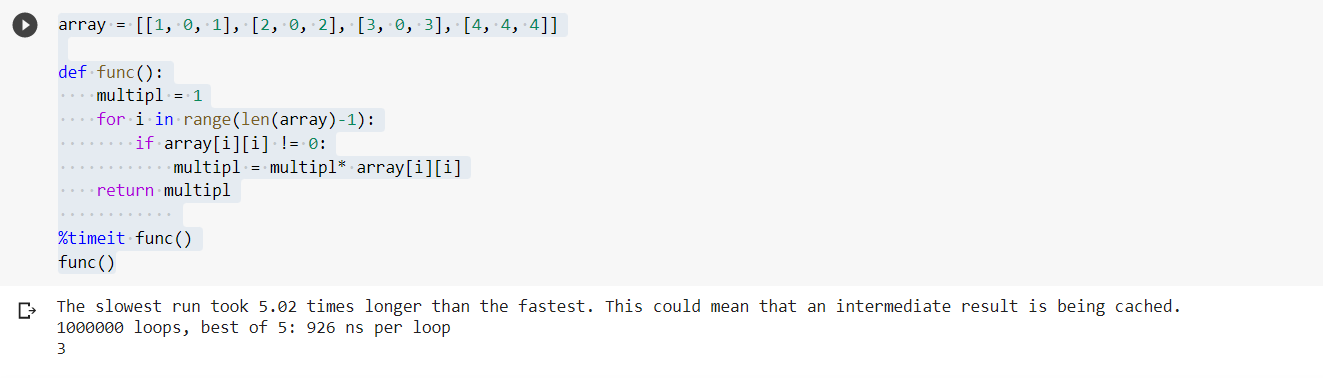
**Задание 1:**

Подсчитать произведение ненулевых элементов на диагонали прямоугольной матрицы.

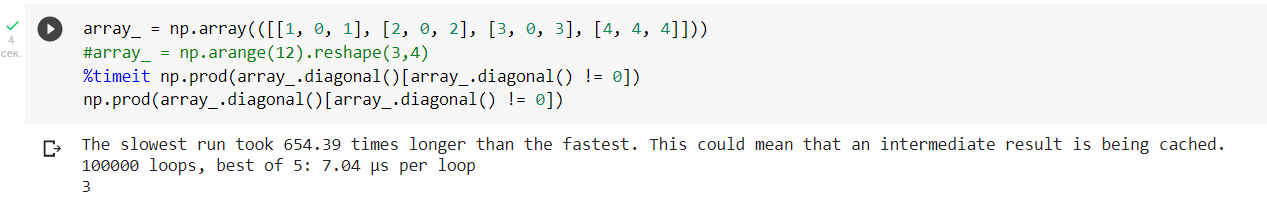
Пример: x = np.array([[1, 0, 1], [2, 0, 2], [3, 0, 3], [4, 4, 4]])

Ответ: 3.

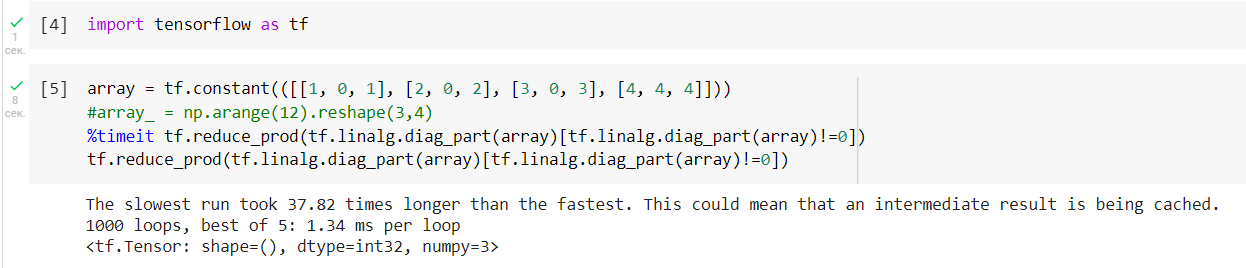
1. Python



1. NumPy



1. Tensorflow



**Задание 2:**

Дана матрица x и два вектора одинаковой длины i и j. Построить вектор np.array([X[i[0], j[0]], X[i[1], j[1]], . . . , X[i[N-1], j[N-1]]]). Пример:

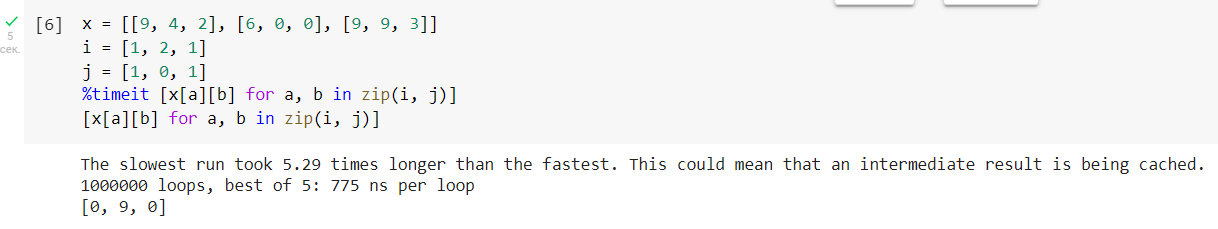
x = [[9 4 2], [6 0 0], [9 9 3]]

i: [1 2 1]

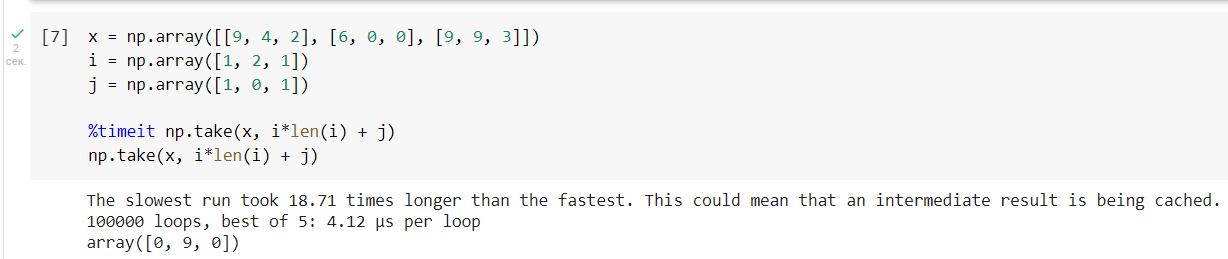
j: [1 0 1]

Ответ: [0 9 0]

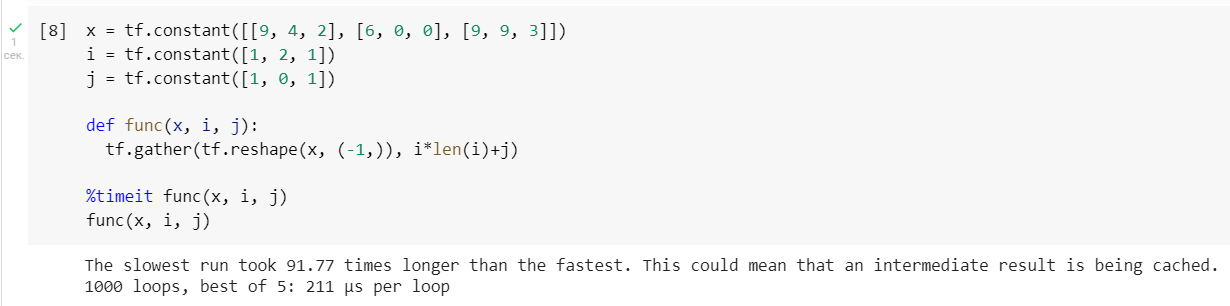
1. Python



1. NumPy



1. Tensorflow



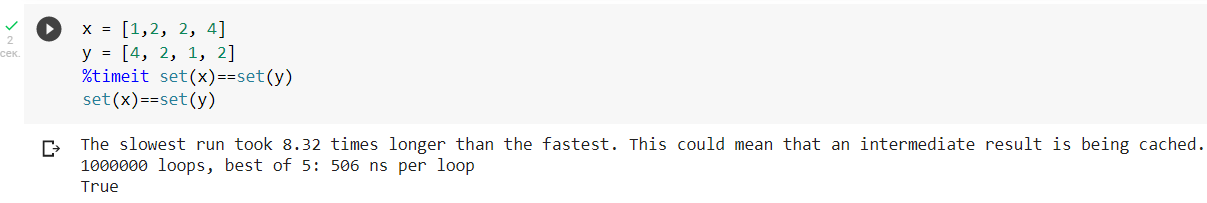
**Задание 3:**

Даны два вектора x и y. Проверить, задают ли они одно и то же мультимножество.

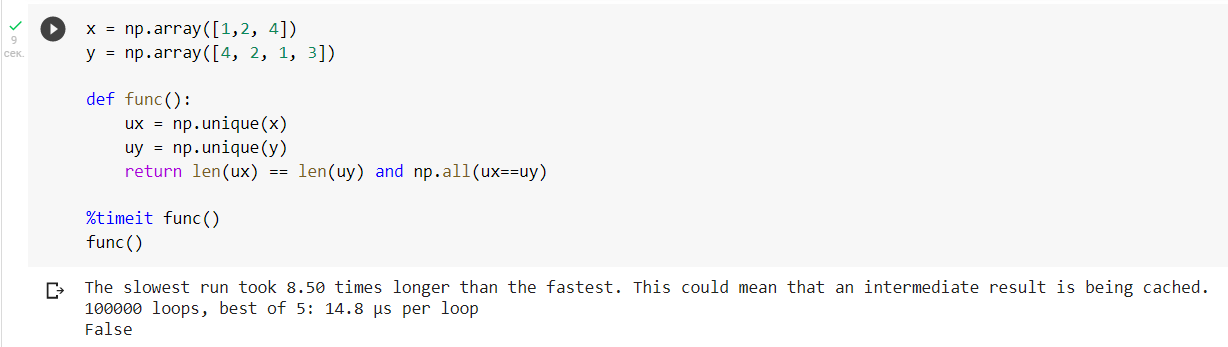
Пример: x = np.array([1,2, 2, 4]), y = np.array([4, 2, 1, 2])

Ответ: True.

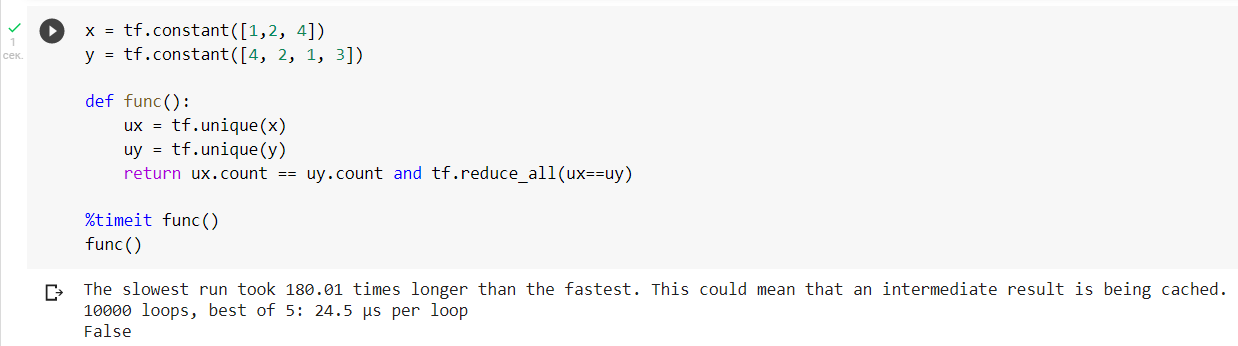
1. Python



1. NumPy



1. Tensorflow



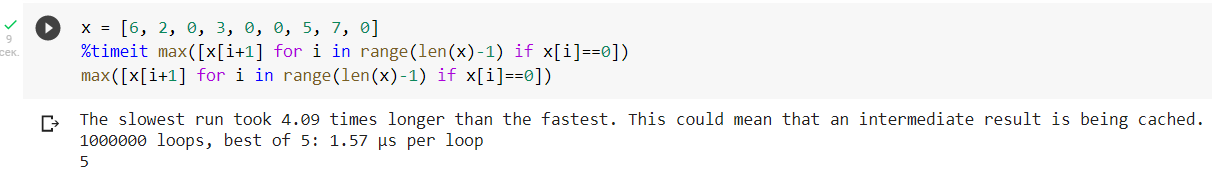
**Задание 4:**

Найти максимальный элемент в векторе x среди элементов, перед которыми стоит нулевой.

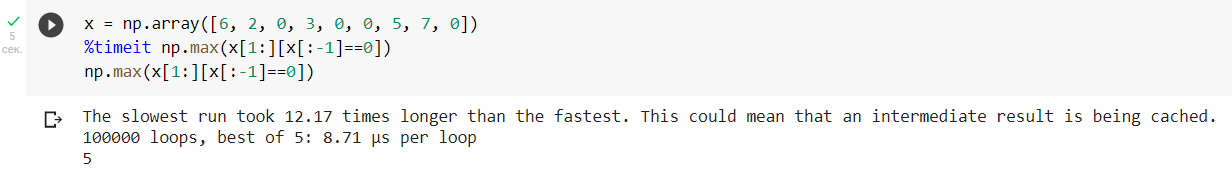
Пример: x = np.array([6, 2, 0, 3, 0, 0, 5, 7, 0])

Ответ: 5.

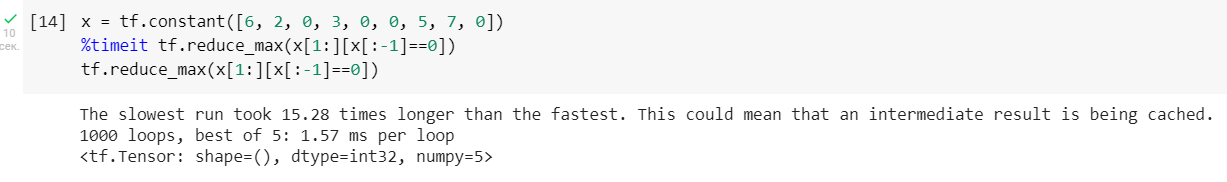
1. Python



1. NumPy



1. Tensorflow



**Задание 5:**

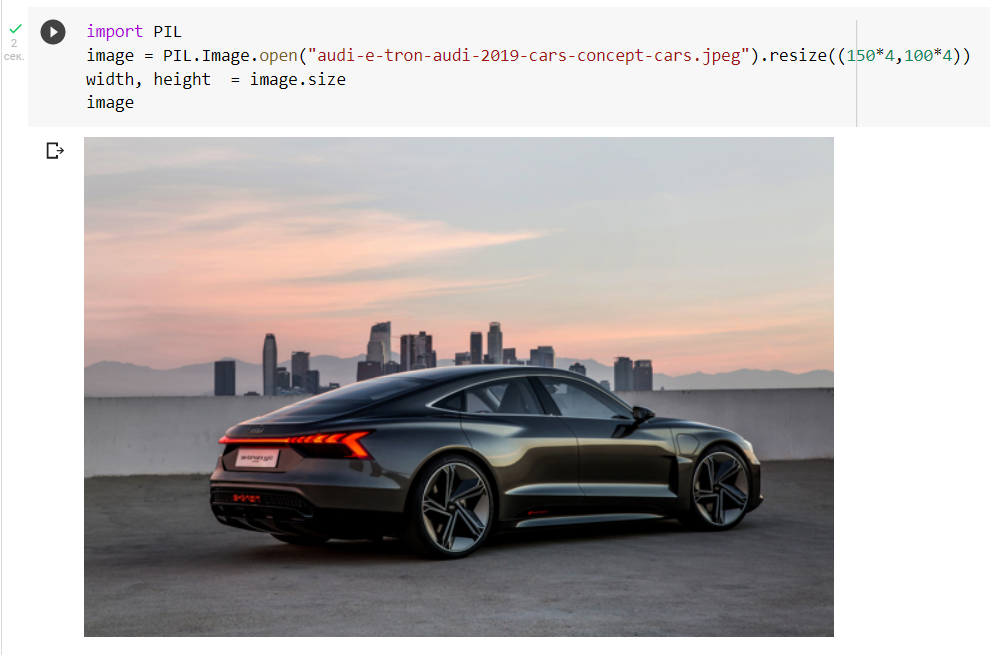
Дан трёхмерный массив, содержащий изображение, размера (height, width, numChannels), а также вектор длины numChannels. Сложить каналы изображения с указанными весами, и вернуть результат в виде матрицы размера (height, width). Считать реальное изображение можно при помощи функции scipy.misc.imread (если изображение не в формате png, установите пакет pillow: condainstallpillow). Преобразуйте цветное изображение в оттенки серого, использовав коэффициенты np.array([0.299, 0.587, 0.114]).

Пример:

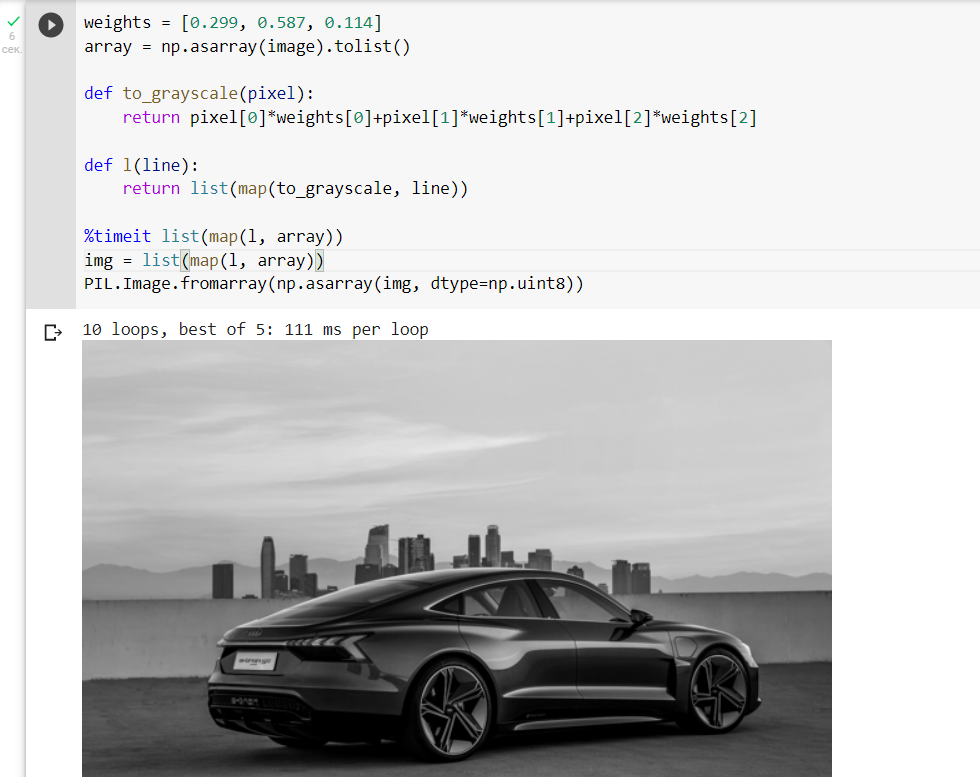


Ответ:

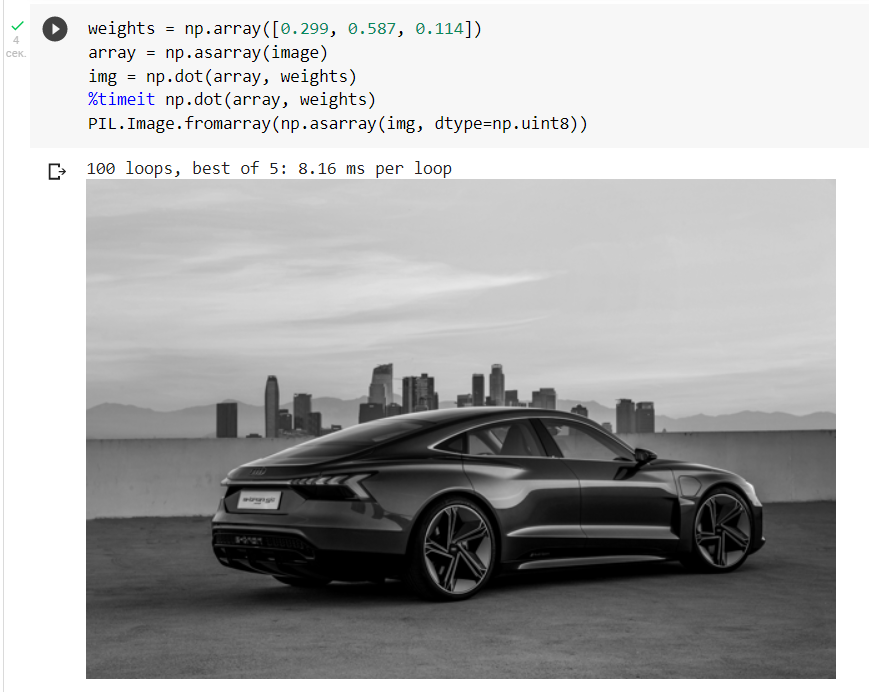




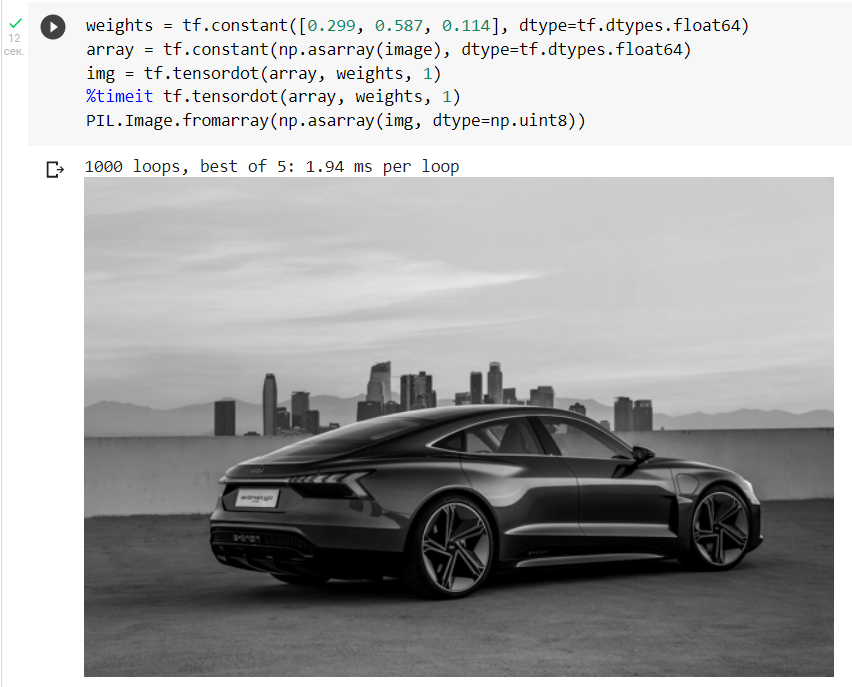
1. Python



1. NumPy



1. Tensorflow



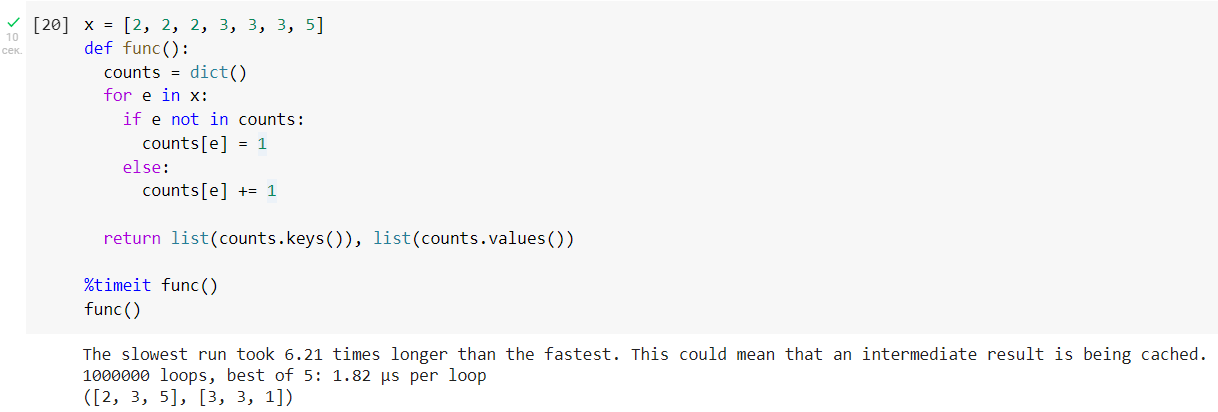
**Задание 6:**

Реализовать кодирование длин серий (Run-lengthencoding). Дан вектор x. Необходимо вернуть кортеж из двух векторов одинаковой длины. Первый содержит числа, а второй - сколько раз их нужно повторить.

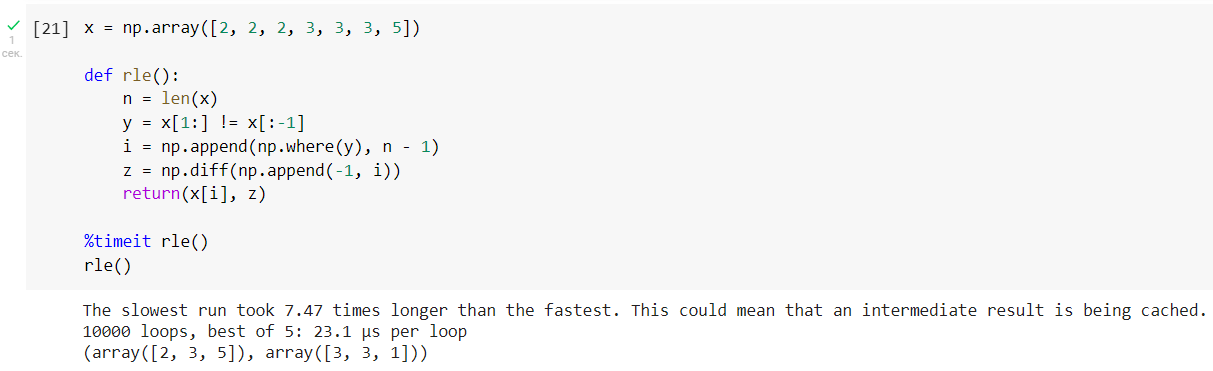
Пример: x = np.array([2, 2, 2, 3, 3, 3, 5]).

Ответ: (np.array([2, 3, 5]), np.array([3, 3, 1])).

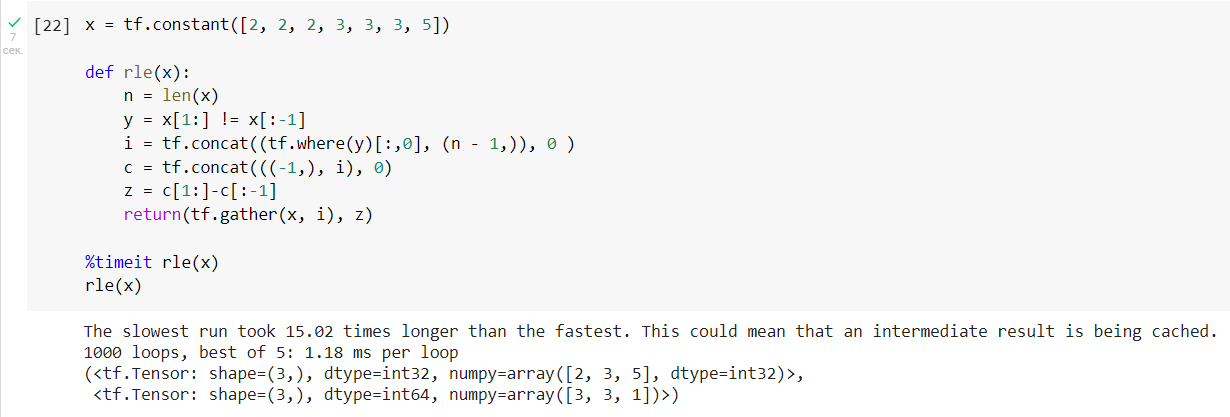
1. Python



1. NumPy



1. Tensorflow



**Задание 7:**

Даны две выборки объектов - X и Y. Вычислить матрицу евклидовых расстояний между объектами. Сравнить с функцией scipy.spatial.distance.euclidean.

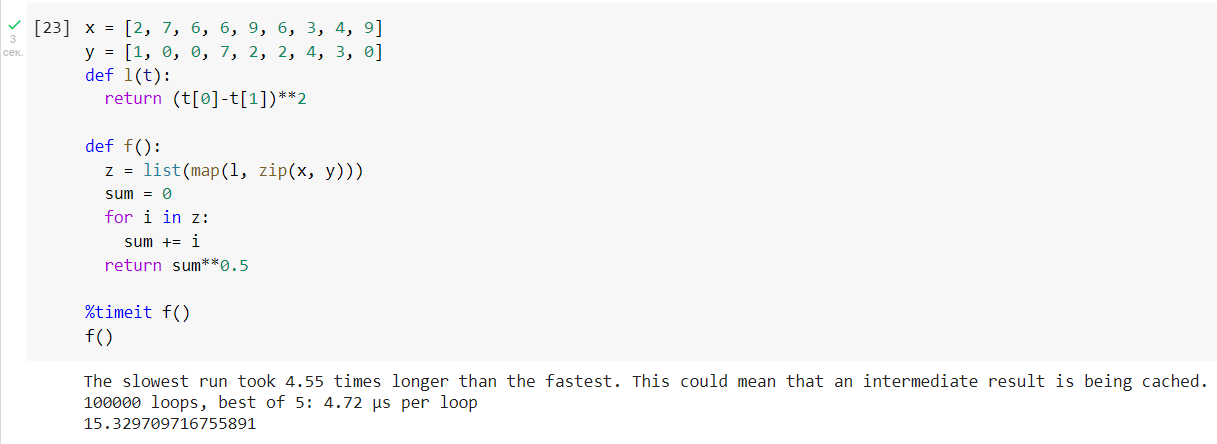
Пример:

x: [2 7 6 6 9 6 3 4 9]

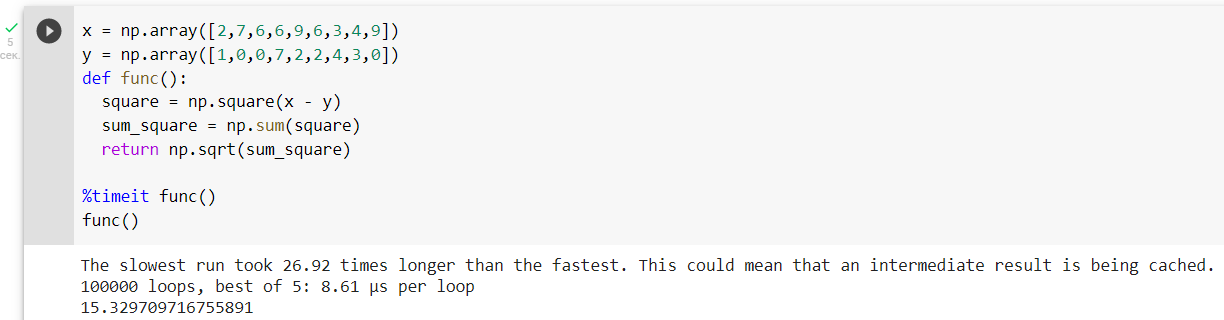
y: [1 0 0 7 2 2 4 3 0]

Ответ: 15.329709716755891

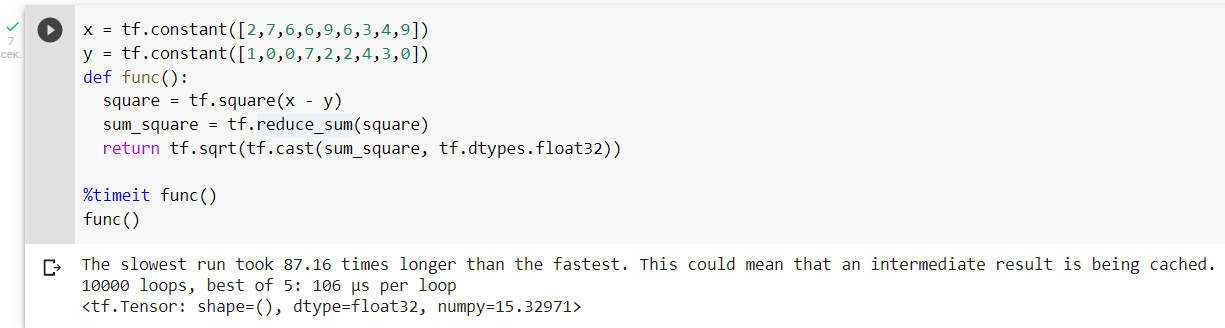
1. Python



1. NumPy



1. Tensorflow



**Задание 8:**

Реализовать функцию вычисления логарифма плотности многомерного нормального распределения. Входные параметры: точки X, размер (N, D), мат. ожидание m, вектор длины D, матрица ковариаций C, размер (D, D). Разрешается использовать библиотечные функции для подсчета определителя матрицы, а также обратной матрицы, в том числе в невекторизованном варианте. Сравнить с scipy.stats.multivariate\_normal(m, C).logpdf(X) как по скорости работы, так и по точности вычислений

1. Python



1. NumPy



1. Pandas



**Вывод:**изучила NumPy. Сравнила производительность NumPy с производительностью обычного Python и с классической библиотекой Tensorflow.