Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Ульяновский государственный технический университет»

Кафедра «Вычислительная техника»

**Отчет по лабораторной работе №2**

Дисциплина: «Разработка профессиональных приложений»

Двумерные структуры данных

Вариант 5

Выполнил:

студент группы ИВТАСбд-22

Зейнетдинов М.М.

Проверил:

преподаватель кафедры

«Вычислительная техника»

Исхаков И.И.

Ульяновск, 2023**Задание по варианту**

Вариант 5:

1. Выполнить обработку элементов прямоугольной матрицы A, имеющей N строк и M столбцов. Определить средние значения по всем строкам и столб- цам матрицы. Результат оформить в виде матрицы из N + 1 строк и M + 1 столбцов.
2. Ввод элементов списка должен быть доступен путем автоматической генерации. Необходимо использовать библиотеку numpy. Результаты выполнения должны сохраняться в файл (исходные данные и результат обработки).
3. Исходный код должен быть откомментирован
4. Необходимо реализовать правильную декомпозицию программы на методы.

**Описание реализации**

Для работы с массивом была использована библиотека numpy, в которой кроме базового варианта (многомерные массивы в базовом варианте) NumPy включает в себя набор пакетов для решения специализированных задач, например:

1. numpy.linalg — реализует операции линейной алгебры (простое умножение векторов и матриц есть в базовом варианте);
2. numpy.random — реализует функции для работы со случайными величинами;
3. numpy.fft — реализует прямое и обратное преобразование Фурье.

Массив создаётся с помощью метода .array(), в атрибутах которого можно указывать какого типа данные будут храниться в массиве, размерность массива, автозаполнение и прочее.

Случайное заполнения массива целыми числами в диапазоне от 0 до 10 выполняется благодаря методу .random.randint(0,10,size=(a,b)). В данном методе создаётся массив с a количеством строк и b количеством столбцов, которые заполняются случайными числами.

Листинг 1. Автоматическое заполнение массива

|  |
| --- |
| arr = np.random.randint(0, 10, size=(a, b)) |

Ручное заполнение массива осуществлено благодаря созданной отдельной функцией, в которую передаются значения количества строк и столбцов.

В данной функции создаётся временный двумерный массив, после чего считываются элементы построчно одной целой строкой, где все элементы разделяются пробелами. Это осуществляется с помощью цикла и методов .array() и .split(). То есть создаётся массив, в котором элементами являются разделённые пробелом числа введённой строки. После чего он записывается в i-тый элемент временного двумерного массива. После обхода всех строк функция возвращает полученный временный массив.

Листинг 2. Функция handinput()

|  |
| --- |
| def handinput(a, b): # функция ввода элементов массива  tempcomarr = np.empty(shape=(a, b), dtype='int64')  for i in range(a):  try:  print("Введите элементы " + str(i) + "-ой строки массива через пробел")  temparr = str(input())  tempcomarr[i] = np.array(list(map(int, temparr.split(' '))))  except ValueError:  print("Некорректный ввод")  exit()  return tempcomarr |

Для реализации основного алгоритма программы были использованы методы .append() .split() и np.mean(). Метод .mean нужен для того чтобы вычислить среднее значения стоки или столбца, это делается с помощью указания axis(0 – столбец, 1-строка). После чего методом .append() в котором для того чтобы добавить новую строку нужно указать также через axis(0) а также т.к. np.mean() возвращает массив, то мы получаем нужную для нас строку. Но для того чтобы добавить новый столбец потребовалось массив полученный в результате выполнения np.mean() разделить методом .split(), который разделяет 1 массив на 3 разных объединяя их в список, т.к. .append() для столбцов требует отдельный элемент для добавления их в конец каждой строки.

Листинг 3. Функция algorythm()

|  |
| --- |
| def algorythm(calcarr): # функция алгоритма  calcarr = np.append(calcarr, np.array([np.mean(calcarr, axis=0)]), 0)  calcarr = np.append(calcarr, np.split(np.array(np.mean(calcarr, axis=1)), len(calcarr)), 1)  return calcarr |

Для вывода массива в отдельный файл была использована функция open(“output.txt”, “w+”) “w+” позволяет нам создать файл при его отсутствии, а также записать в него новые данные. Через метод .write() осуществляется сам ввод в файл. После чего методом .close() файл закрывается.

**Описание возникших затруднений**

В данной лабораторной работе возникли трудности с изучением различных методов из библиотеки numpy. Поскольку методов не малое количество, а также каждый метод является особенным.

**Описание альтернативных способов решения**

Алгоритм можно было реализовать немного иначе, с помощью применения списков вместо массивов, для которых бы потребовалось написать свой алгоритм нахождения среднего значения строки и столбца. Также массив можно было считывать с файла, для этого есть специально предусмотренный метод .load() из библиотеки numpy.

Тестирование

|  |  |
| --- | --- |
|  |  |

Рис 1. Автоматический ввод массива

|  |  |
| --- | --- |
|  |  |

Рис 2. Ручной ввод массива

Вывод

В данной лабораторной работе были приобретены навыки владения библиотекой numpy, знания использования двумерных массивов на языке Python а также знания использования файлов. В результате выполнения была получена программа, которая считывает двумерный массив, после чего в дополнительный отведённый столбец и строку записываются средние значения столбцов и строк.

**Код программы**

import numpy as np  
  
def myinput(): # функция ввода числовых значений  
 try:  
 val = int(input())  
 except ValueError:  
 print("Некорректный ввод")  
 exit()  
 return val  
  
  
def algorythm(calcarr): # функция алгоритма  
 calcarr = np.append(calcarr, np.array([np.mean(calcarr, axis=0)]), 0)  
 calcarr = np.append(calcarr, np.split(np.array(np.mean(calcarr, axis=1)), len(calcarr)), 1)  
 return calcarr  
  
  
def handinput(a, b): # функция ввода элементов массива  
 tempcomarr = np.empty(shape=(a, b), dtype='int64')  
 for i in range(a):  
 try:  
 print("Введите элементы " + str(i) + "-ой строки массива через пробел")  
 temparr = str(input())  
 tempcomarr[i] = np.array(list(map(int, temparr.split(' '))))  
 except ValueError:  
 print("Некорректный ввод")  
 exit()  
 return tempcomarr  
  
def main(): # функция выполнения программы  
  
 print("Введите:\n1-чтобы ввести данные самому\n0-чтобы ввести случайные данные")  
 variant = myinput()  
  
 print("Введите количество строк массива")  
 a = myinput()  
 print("Введите количество столбцов массива")  
 b = myinput()  
  
 if variant == 1:  
 arr = handinput(a, b)  
 elif variant == 0:  
 arr = np.random.randint(0, 10, size=(a, b)) # заполнение массива случайными значениями от 0 до 10  
 else:  
 print("Некорректный ввод")  
 exit()  
  
 print("Исходный массив A[" + str(a) + "][" + str(b) + "]=\n" + str(arr))  
 calcarr = np.copy(arr)  
 calcarr = algorythm(calcarr)  
 print("Обработанный массив B[" + str(a + 1) + "][" + str(b + 1) + "]=\n" + str(calcarr))  
 f = open("output.txt", "w+") # открывание(при отсутствии создание) файла для записи  
 try:  
 f.write("Исходный массив A[" + str(a) + "][" + str(b) + "]=\n" + str(arr) + '\n') # запись результатов  
 f.write("Обработанный массив B[" + str(a + 1) + "][" + str(b + 1) + "]=\n" + str(calcarr) + '\n')  
 finally: # нужен для срабатывания f.close() при любом типе исключений  
 f.close()  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()