

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Лабораторная работа №1
по дисциплине
“Линейная алгебра и анализ данных”

Семестр I

Выполнили:
студенты

Тиганов Вадим Игоревич
гр. J3112
ИСУ 467701

Вагин Арсений Антонович
гр. J3112
ИСУ 465339

Отчет сдан:
XX.12.2024

Санкт-Петербург
2024

Цель лабораторной работы

Освоить основные концепции линейной алгебры и анализа данных по работе с матрицами. Научиться реализовывать и тестировать алгоритмы работы с матрицами в разреженно-строчном формате. Изучить и понять принципы работы алгоритмов, а также .tex верстания для создания отчета.

Задачи лабораторной работы

1. Реализация хранения матриц в разреженно-строчном виде.
2. Реализация операций над матрицами.
3. Вычисление определителя и проверка существования обратной матрицы.
4. Тестирование и проверка правильности работы алгоритмов.
5. Верстка отчета в формате \LaTeX

Ход выполнения лабораторной работы

Задача 1

Задача заключалась в реализации следующих функций в классе: (был выбран ЯП Python, полный листинг кода см. в приложении А)

- Ввод матрицы заданного размера пользователем.
- Подсчет следа матрицы.
- Поиск и вывод элемента матрицы по заданным индексам.
- Тестирование работы программы и создание консольного пользовательского интерфейса.

Использованные библиотеки и инструменты языка

В ходе написания программы были использованы только стандартные средства языка. Также для удобства и лучшей читаемости кода была импортирована библиотека `typing`.

Реализация функций и основные идеи

Все функции были реализованы в классе `MatrixKeeper`

Написаны функции:

`inputMatrix` - ввод матрицы пользователем,

`trace` - поиск следа матрицы,

`findByIndex` - поиск элемента по введенному индексу.

Суть работы алгоритмов:

- `inputMatrix`: Приглашение пользователя ко вводу. Вначале через пробел вводятся два числа типа `int` - размер матрицы. Вторым приглашением вводится матрица по строке, элементы в строке разделяются пробелом.
- `trace`: След матрицы - сумма элементов главной диагонали этой матрицы. Циклом, оставаясь в пределах матрицы, проходимся по элементам с индексами вида `[i][i]`, считаем сумму таких элементов. Можем так делать по той причине, что матрица имеет следующую структуру в классе:

```
1 self.matrix: Optional[List[List[float]]]
```

- то есть храним матрицу как список, каждый элемент которого является тоже списком.

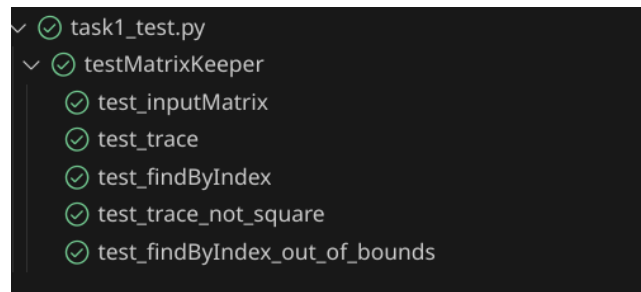
- `findByIndex`: возвращаем элемент из матрицы, отнимая от индексов по единице, т.к. в ЯП отсчет начинается с нуля.

```
1 return self.matrix[n-1][m-1]
```

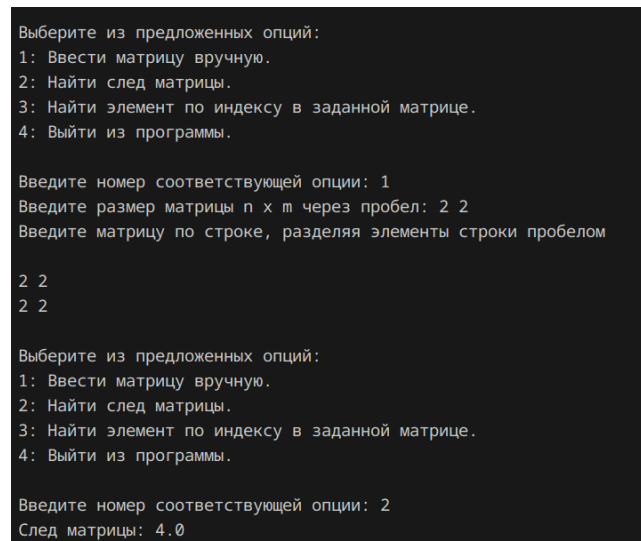
Тестирование программы

Написаны юниттесты для каждой функции класса с помощью стандартной библиотеки `unittest`. Также протестировано вручную.

Листинг кода теста см. в приложении A-test.



Успешное прохождение unittests



Ручной тест поиска следа матрицы 2x2 со всеми элементами, равными 2.

```

Введите номер соответствующей опции: 1
Введите размер матрицы n x m через пробел: 9 5
Введите матрицу по строке, разделяя элементы строки пробелом

1 2 3 5 1
1 6 9 2 4
3 5 1 2 5
1 2 5 7 4
-10000 1 6 32 1
0.0003 1 2 6 3
4 2 8 3 7
5 4 2 3 6
5 3 1 2 7

Выберите из предложенных опций:
1: Ввести матрицу вручную.
2: Найти след матрицы.
3: Найти элемент по индексу в заданной матрице.
4: Выйти из программы.

Введите номер соответствующей опции: 3
Введите номер строки: 5
Введите номер столбца: 1
Элемент матрицы [5][1] = -10000.0

```

Ручной тест поиска страшного элемента в страшной матрице 9x5.

Итак, справились с первым заданием.

Задача 2

Во второй задаче требуется реализовать три функции для операций с матрицами: (полный листинг кода см. в приложении Б)

- Сложение двух матриц.
- Умножение двух матриц.
- Умножение матрицы на скаляр.

Использованные библиотеки и инструменты языка

В ходе написания программы были использованы только стандартные средства языка. Также для удобства и лучшей читаемости кода была импортирована библиотека `typing`.

Были реализованы 3 функции:

`matrixAddition` - сложение двух матриц.

`matrixByMatrixMultiplication` - перемножение двух матриц.

`matrixScalarMultiplication` - умножение одной из двух матриц на заданное число.

Согласно техническому заданию, функция ввода матрицы пользователем была импортирована из файла предыдущего задания. (`inputMatrix`)

Суть работы алгоритмов:

- `matrixAddition`: Классическое сложение матрицы. Возвращаем матрицу, где каждый элемент с определенными индексами равен сумме элементов с соответствующими индексами из складываемых матриц.

$$(A + B)_{i,k} = A_{i,k} + B_{i,k}$$

- `matrixByMatrixMultiplication`: Умножение матрицы на матрицу. Возвращаем матрицу, где каждый элемент с определенными индексами равен сумме произведений элементов соответствующей строки первой матрицы и столбца второй матрицы.

$$(AB)_{i,j} = \sum_{k=1}^n A_{i,k} \cdot B_{k,j}$$

- `matrixScalarMultiplication`: Умножение матрицы на скаляр. Возвращаем матрицу, где каждый элемент равен произведению соответствующего элемента исходной матрицы и скаляра.

$$(cA)_{i,j} = c \cdot A_{i,j}$$

Тестирование программы

Написаны юниттесты для каждой функции класса с помощью стандартной библиотеки `unittest`. Также протестировано вручную.

Листинг кода теста см. в приложении Б-test.

```
✓ task2_test.py
  ✓ TestMatrixOperations
    ✓ test_matrixAddition
    ✓ test_matrixByMatrixMultiplication
    ✓ test_matrixScalarMultiplication
```

Успешное прохождение unittests

```

Введите размер матрицы n x m через пробел: 2 2
Введите матрицу по строке, разделяя элементы строки пробелом

1 2
3 4

Выберите из предложенных опций:
1: Ввести первую матрицу вручную.
2: Ввести вторую матрицу вручную.
3: Сложить две матрицы.
4: Умножить матрицу на матрицу.
5: Умножить матрицу на скаляр.
6: Выйти из программы.

Введите номер соответствующей опции: 2
Введите размер матрицы n x m через пробел: 2 2
Введите матрицу по строке, разделяя элементы строки пробелом

6 5
7 9

Выберите из предложенных опций:
1: Ввести первую матрицу вручную.
2: Ввести вторую матрицу вручную.
3: Сложить две матрицы.
4: Умножить матрицу на матрицу.
5: Умножить матрицу на скаляр.
6: Выйти из программы.

Введите номер соответствующей опции: 3
Результат сложения матриц:
[7.0, 7.0]
[10.0, 13.0]

```

Ручной тест сложения двух матриц 2 на 2.

```

Введите номер соответствующей опции: 1
Введите размер матрицы n x m через пробел: 2 2
Введите матрицу по строке, разделяя элементы строки пробелом

1 2
6 7

Выберите из предложенных опций:
1: Ввести первую матрицу вручную.
2: Ввести вторую матрицу вручную.
3: Сложить две матрицы.
4: Умножить матрицу на матрицу.
5: Умножить матрицу на скаляр.
6: Выйти из программы.

Введите номер соответствующей опции: 5
Выберите матрицу для умножения на скаляр (1 - первая матрица, 2 - вторая матрица): 1
Введите скаляр: 2.718281828459045
Результат умножения матрицы на скаляр:
[2.718281828459045, 5.43656365691809]
[16.30969097075427, 19.027972799213316]

```

Ручной тест умножения матрицы 2 на 2 на число Эйлера с небольшим количеством знаков после запятой.

```
Введите номер соответствующей опции: 1
Введите размер матрицы n x m через пробел: 2 2
Введите матрицу по строке, разделяя элементы строки пробелом

1 6
-0.1238 52

Выберите из предложенных опций:
1: Ввести первую матрицу вручную.
2: Ввести вторую матрицу вручную.
3: Сложить две матрицы.
4: Умножить матрицу на матрицу.
5: Умножить матрицу на скаляр.
6: Выйти из программы.

Введите номер соответствующей опции: 2
Введите размер матрицы n x m через пробел: 2 2
Введите матрицу по строке, разделяя элементы строки пробелом

0.312837 2
1 1

Выберите из предложенных опций:
1: Ввести первую матрицу вручную.
2: Ввести вторую матрицу вручную.
3: Сложить две матрицы.
4: Умножить матрицу на матрицу.
5: Умножить матрицу на скаляр.
6: Выйти из программы.

Введите номер соответствующей опции: 4
Результат умножения матриц:
[6.312837, 8.0]
[51.9612707794, 51.7524]
```

Ручной тест умножения двух страшных матриц 2 на 2.

Matrix Calculator

Matrix A Input		Matrix B Input	
row	column	row	column
2	2	2	2
1	6	.12837	2
-0.123	52	1	1
Clear All 0 All 1 Random		Clear All 0 All 1 Random	
Transpose Power of 2		Transpose Power of 2	
Determinant Inverse × 3		Determinant Inverse × 3	
A + B A - B AB A ↔ B			
Result			
$AB = \begin{bmatrix} 6.312837 & 8 \\ 51.96127078 & 51.7524 \end{bmatrix}$			

Удостоверимся в правильности полученного результата, умножив матрицы на сайте-калькуляторе. Результат правильный.

Справились со вторым заданием.

Вывод

Приложения

Список использованной литературы