

	<p align="center"> Министерство образования и науки Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана) </p>
---	--

ФАКУЛЬТЕТ _____ Информатика и системы управления
 КАФЕДРА _____ Программное обеспечение ЭВМ и информационные технологии

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ 3
По дисциплине «Типы и структуре данных»

Название **«Обработка разреженных матриц»**

Студент Дубов Андрей Игоревич
 фамилия, имя, отчество

Группа ИУ7-33Б

Вариант 5

Тип лабораторной работы Учебная

Студент	_____	<u>Дубов А. И.</u>
	<i>подпись, дата</i>	<i>фамилия, и.о.</i>
Преподаватель	_____	<u>Рыбкин Ю. А.</u>
	<i>подпись, дата</i>	<u>Силантьева А. В.</u>
		<i>фамилия, и.о.</i>

2022 г.

Оглавление

Условие задачи.....	3
Описание технического задания.....	3
Входные данные:.....	3
Либо файл, либо ручной ввод матрицы в разреженном виде.....	3
Выходные данные:.....	3
Аварийные ситуации:.....	3
Описание структуры данных.....	3
Описание алгоритма.....	4
Оценка эффективности.....	5
Вывод.....	7
Ответы на контрольные вопросы.....	7

Условие задачи

Разработать программу умножения или сложения разреженных матриц. Предусмотреть возможность ввода данных, как с клавиатуры, так и использования заранее подготовленных данных. Матрицы хранятся и выводятся в форме трех объектов. Для небольших матриц можно дополнительно вывести матрицу в виде матрицы. Величина матриц - любая (допустим, 1000×1000). Сравнить эффективность (по памяти и по времени выполнения) стандартных алгоритмов обработки матриц с алгоритмами обработки разреженных матриц при различной степени разреженности матриц и различной размерности матриц.

Описание технического задания

Разреженная (содержащая много нулей) матрица хранится в форме 3-х объектов:

- вектор A содержит значения ненулевых элементов;
- вектор JA содержит номера столбцов для элементов вектора A;
- связный список IA, в элементе N_k которого находится номер компонент в A и JA, с которых начинается описание строки N_k матрицы A.

1. Смоделировать операцию умножения вектора-строки и матрицы, хранящихся в этой форме, с получением результата в той же форме.
2. Произвести операцию умножения, применяя стандартный алгоритм работы с матрицами.
3. Сравнить время выполнения операций и объем памяти при использовании этих 2-х алгоритмов при различном проценте заполнения матриц.

Входные данные:

Вводятся размеры матрицы, потом вводится по строкам. Сначала вводится количество элементов в строке, потом вводится по два значения, элемент и номер колонны элемента.

Выходные данные:

1. Матрица в разреженном виде, вектор в разреженном виде, они же в нормальном виде и результаты их перемножения

Аварийные ситуации:

1. Некорректный ввод чисел

Описание структуры данных

Структура разреженной матрицы

```
typedef struct
{
    long long nums[MAX_NUMBERS];
```

```

    long long rows[MAX_NUMBERS];
    long long columns[MAX_SIDE_LEN];
    int n;
    int m;
    int size;
} sparse_matrix_t;

```

nums – ненулевые элементы

rows – номера рядов этих элементов

columns – массив начала каждой колонны

n – количество колонн

m – количество строк

size – количество чисел

Структура разреженного вектора

```

typedef struct
{
    long long nums[MAX_NUMBERS];
    long long columns[MAX_SIDE_LEN];
    int n;
    int size;
} sparse_vector_t;

```

nums – ненулевые элементы

columns – номера рядов этих элементов

n – количество элементов

size – количество чисел

Структура матрицы

```

typedef struct
{
    long long nums[MAX_SIDE_LEN][MAX_SIDE_LEN];
    int n;
    int m;
} matrix_t;

```

nums – элементы

n – количество колонн

m – количество строк

Структура вектора

```

typedef struct
{
    long long nums[MAX_NUMBERS];
    int n;
} vector_t;

```

nums – элементы

n – количество элементов

Описание алгоритма

Программа проит по массиву чисел и последовательно складывает нужное произведение в требуемый столбец ветора ответа.

Оценка эффективности

Время работы, все времена измеряются в тиках процессора

	Добавление	Удаление
Список	7	7
Массив	18	19

1% заполнения

	Разреженная матрица	Обычная матрица
10x10	245	1188
100x100	33083	79338
200x200	358851	314572
300x300	1396463	1192606

2% заполнения

	Разреженная матрица	Обычная матрица
10x10	1067	1156
100x100	106313	87527
200x200	1090175	317017
300x300	3352081	908417

3% заполнения

	Разреженная матрица	Обычная матрица
10x10	828	1114
100x100	175832	79680
200x200	1502678	368860
300x300	4943702	956393

4% заполнения

	Разреженная матрица	Обычная матрица
10x10	1063	1156
100x100	254472	78362
200x200	2088345	351017
300x300	6790451	996616

6% заполнения

	Разреженная матрица	Обычная матрица
10x10	1275	1075

100x100	420381	78786
200x200	3116522	343669
300x300	10345181	1132095

8%

	Разреженная матрица	Обычная матрица
10x10	1529	1075
100x100	551683	78743
200x200	4410930	350596
300x300	14244226	1164198

9% заполнения

	Разреженная матрица	Обычная матрица
10x10	1957	1120
100x100	647717	78416
200x200	5075631	316805
300x300	16595531	738796

10% заполнения

	Разреженная матрица	Обычная матрица
10x10	2158	1045
100x100	2398253	226246
200x200	5589672	316353
300x300	18708886	743819

20% заполнения

	Разреженная матрица	Обычная матрица
10x10	4541	1276
100x100	1595300	79334
200x200	12034178	318805
300x300	40098071	772540

30% заполнения

	Разреженная матрица	Обычная матрица
10x10	6003	1116
100x100	2547311	94232
200x200	19171619	314299
300x300	61574826	1006737

Вывод

Использование алгоритмов хранения и обработки разреженных матриц выгодно при маленьком количестве элементов, примерно до 1-2% заполненности матриц. В таком случае, алгоритм выигрывает как и в размерах занимаемой памяти, так и в скорости обработки. Но при заполненности более чем 2%, алгоритм обработки и хранения разреженных матриц начинает проигрывать по времени. Стоит отметить, что для каждого значения в разреженной матрице хранится по сути два числа, то есть пренебрегая массивом, которые указывают на начала строк, то до 50% разреженная матрица выигрывает по памяти.

Ответы на контрольные вопросы

1. Что такое разреженная матрица, какие способы хранения вы знаете?

Разреженная матрица – это матрица, содержащая большое количество нулей. Способы хранения: связная схема хранения, строчный формат, линейный связный список, кольцевой связный список, двунаправленные стеки и очереди.

2. Каким образом и сколько памяти выделяется под хранение разреженной и обычной матрицы?

Под обычную матрицу выделяет $N * M$ ячеек памяти, где N – строки, а M – столбцы. Для разреженной матрицы – зависит от способа. В случае разреженного формата, требуется $2 * K + N$ ячеек памяти, где K – количество ненулевых элементов.

3. Каков принцип обработки разреженной матрицы?

Алгоритмы обработки разреженных матриц предусматривают действие только с ненулевыми элементами, и, таким образом, количество операций будет пропорционально количеству ненулевых элементов.

4. В каком случае для матриц эффективнее применять стандартные алгоритмы обработки матриц? От чего это зависит?

Стандартные алгоритмы обработки матриц эффективнее применять при большом количестве ненулевых элементов (от 50%).