



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)

ФАКУЛЬТЕТ

ИУ

КАФЕДРА

ИУ7

ОТЧЕТ ПО ЛР3 ТИПОВ И СТРУКТУР ДАННЫХ, **Вариант 3**

Студент

Андреев Александр Алексеевич
фамилия, имя, отчество

Группа

ИУ7-34Б

Тип практики

учебная

Студент

подпись, дата

фамилия, и.о.

Преподаватель

подпись, дата

фамилия, и.о.

Оценка

9 октября 2020

Оглавление

Оглавление	1
1. Цель работы	2
2. Описание условия задачи	2
3. Описание ТЗ, включающее внешнюю спецификацию	2
а. Описание исходных данных	2
б. Описание задачи, реализуемой программой	2
с. Способ обращения программы	3
д. Описание возможных аварийных ситуаций и ошибок пользователя	5
4. Описание внутренних СД	6
5. Алгоритм	6
6. Набор тестов с указанием, что проверяется	8
7. Выводы по проделанной работе	11
8. Контрольные вопросы	16

1. Цель работы

В качестве цели работы ставится реализация алгоритма обработки разреженных матриц, сравнение эффективности использования этих алгоритмов по времени выполнения и требуемой памяти со стандартными алгоритмами обработки матриц при различном процентном заполнении матриц ненулевыми значениями и при различных размерах матриц.

2. Описание условия задачи

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.

3. Описание ТЗ, включающее внешнюю спецификацию

а. Описание исходных данных

Для каждой операции указано, как задавать к ней параметры, а также будет ли требоваться ручной ввод из консоли (и предложено указать формат - как в системе или классический) или же значения будут сгенерированы системой.

Ограничения на вводимые данные:

- Значения по умолчанию все элементы матриц представлены в виде 4-байтового целого
- Значения для размеров только положительны
- Для процентной разреженности - значения неотрицательны

б. Описание задачи, реализуемой программой

Программа имеет два уровня меню, где первое - Основное, второе - Меню операций.

В основном меню программа должна предложить пользователю на выбор ввод данных через терминал, из текстового файла и посредством генерации.

Рис. 1 (Основное меню программы)

Основное меню программы:

- 1 - Ручной ввод матрицы
- 2 - Чтение матрицы из файла
- 3 - Генерация матрицы
- 4 - Вывод отчетной таблицы сравнения

Укажите пункт меню (1-3):

После выбора соответствующего пункта Основного меню и ввода матрица Пользователь должен получить возможность обращаться с Меню операций.

Рис. 2 (Меню операций программы)

Меню операций:

- 1 - Умножение вектора-строки на матрицу, хранящихся в разреженной форме, с получением результата в той же форме
- 2 - Умножение стандартным алгоритмом работы с матрицами
- 3 - Сравнение времени выполнения операций, объема памяти при использовании этих алгоритмов при различной доли заполненности матриц
- 4 - Вывод в разреженной форме
- 5 - Вывод в простой форме
- 0 - Завершение программы

Укажите пункт меню (0-5):

с. Способ обращения программы

Скомпилированная программа запускается командой “./a.out X” на Unix и “./a.exe X”.

После запуска пользователем программа должна вывести инструкцию по своему применению, Основное меню и приглашение к выбору пункта меню (Рис. 3).

Рис. 3

Программа умножения вектора-строки на матрицу,
хранящихся в разреженной форме, с получением результата в той же форме.

Основное меню программы:

- 1 - Ручной ввод матрицы
- 2 - Чтение матрицы из файла

4 - Вывод отчетной таблицы сравнения

Укажите пункт меню (1-3):

Далее после ввода соответствующей команды программа должна выполнять команды пользователя.

- **Комада “1”**

При указании пункта меню “1” Пользователь получает возможность ввести матрицу вручную.

- **Команда “2”**

При указании пункта меню “2” Пользователь получает возможность считать матрицу из файла.

- **Команда “3”**

При указании пункта меню “3” Пользователь получает возможность считать сгенерировать матрицу заданного размера.

- **Команда “4”**

При указании пункта меню “4” Пользователь получает возможность вывести отчетную таблицу сравнения.

В случае ошибки ввода Пользователя при указании пункта Основного меню Программа должна сообщить об этом.

Далее после ввода соответствующей команды программа должна выполнять команды пользователя.

Рис. 4

Меню операций:

- 1 - Умножение вектора-строки на матрицу, хранящихся в разреженной форме, с получением результата в той же форме
- 2 - Умножение стандартным алгоритмом работы с матрицами
- 3 - Сравнение времени выполнения операций, объема памяти при использовании этих алгоритмов при различной доли заполненности матриц
- 4 - Вывод в разреженной форме
- 5 - Вывод в простой форме
- 0 - Завершение программы

Укажите пункт меню (0-5):

Далее после ввода соответствующей команды программа должна выполнять команды пользователя.

- **Команда “0”**
При указании пункта меню “0” Пользователем программа должна завершить свою работу.
- **Команда “1”**
При указании пункта меню “1” Пользователем программа должна предложить пользователю ввод матрицы, хранящуюся в разреженной форме, с дальнейшим умножением и выводом.
- **Команда “2”**
При указании пункта меню “1” Пользователем программа должна предложить пользователю ввод матрицы с дальнейшим умножением и выводом стандартным способом.
- **Команда “3”**
При указании пункта меню “3” Пользователем программа должна запросить у пользователя заполненность матрицы и вывести сравнительную информацию по обоим способам.
- **Команда “4”**
При указании пункта меню “1” Пользователем программа должна вывести на экран матрицу в разреженной форме.
- **Команда “5”**
При указании пункта меню “1” Пользователем программа должна вывести на экран матрицу в простой форме.

При наличии аварийных ситуаций и ошибок пользователя программа должна вывести соответствующее сообщение и не должна завершить свою работу абортно.

d. Описание возможных аварийных ситуаций и ошибок пользователя

В представленной ниже **Таблице 1** отражены действия программы при различных возможных, допущенных пользователем, при ее использовании ошибках.

Табл. 1

Действие программы при различных ошибках		
№	Действие пользователя	Реакция программы
1	Введен несуществующий номер или строка пункта меню	Программа выведет сообщение об ошибке “Возникла ошибка при вводе. Повторите попытку...”
2	Введено нецелое значение, как элемент матрицы, в ручном режиме или считывании из файла	Программа выведет сообщение об ошибке “Возникла ошибка при вводе. Повторите попытку...”

3	Введен некорректный индекс при заполнении матрицы в ручном режиме или считывании из файла	Программа выведет сообщение об ошибке “Указан неверный размер. Повторите попытку...”
4	Пользователь умножает матрицу стандартным способом, где все элементы нулевые	Программа выведет сообщение об ошибке “В матрице все элементы нулевые”

4. Описание внутренних СД

Считанные из консоли числа внутри программы хранятся в виде структуры **number**, которая представляет из себя (Рис. 1)

```
INPUT_STRING_MAX_SIZE = 256;
```

```
INPUT_TABLE_INFORMATION_MAX_SIZE = 256;
```

Рис. 5

```
struct ja
{
    int num;
    struct ja *next;
};
```

5. Алгоритм

Основная программа

Считав матрицу и находясь в меню операций программы, программа до получения продолжает свою работу до получения от пользователя указаний о выходе из нее.

Выбор действия внутри меню операций:

- **Команда “0”**. При указании пункта меню “0” Пользователем программа должна завершить свою работу.
- **Команда “1”**. При указании пункта меню “1” Пользователем программа должна предложить пользователю ввод матрицы, хранящуюся в разреженной форме, с дальнейшим умножением и выводом.
- **Команда “2”**. При указании пункта меню “1” Пользователем программа должна предложить пользователю ввод матрицы с дальнейшим умножением и выводом стандартным способом.
- **Команда “3”**. При указании пункта меню “3” Пользователем программа должна запросить у пользователя заполненность матрицы и вывести сравнительную информацию по обоим способам.
- **Команда “4”**. При указании пункта меню “1” Пользователем программа должна вывести на экран матрицу в разреженной форме.

- **Команда “5”.** При указании пункта меню “1” Пользователем программа должна вывести на экран матрицу в простой форме.

Описание основных функций:

int mult_std_matrix(int **matrix, int size_rows, int size_cols, int c);

Функция, производящая операцию умножения матрицы, хранящейся в стандартном виде.

int mult_matrix(int *A, int *IA, int*JA, int count, int size, int c);

Функция умножения матрицы, хранящейся в разреженном виде.

int ** manual_input(int *size_rows, int *size_cols, int *count);

Функция ручного ввода матрицы. Здесь пользователь вводит сначала размерность матрицы, после чего указывает количество ненулевых элементов, далее по позициям вводит значения ненулевых элементов

int **input_from_file(int *size_cols, int *size_rows, int *count);

Чтение матрицы из файла.

void output_std_matrix(int **mtr, int size_rows, int size_cols);

Вывод матрицы, хранящейся в стандартном виде, на экран.

void output_matrix(int *A, int *IA, int*JA, int count, int size_cols);

Вывод матрицы, хранящейся в разреженном виде, на экран.

void add_elem(struct ja **JA)

Добавление элементов в матрицу, хранящуюся в разреженном виде.

6. Набор тестов с указанием, что проверяется

В представленных ниже **Таблица 2** отражены тестирование устойчивости работы консольного меню программы и демонстрация устойчивости работы программы к различному типу выполняемых с ней операций пользователем соответственно.

Табл. 2

Тестирование устойчивости работы программы			
№	Ввод пользователя	Реакция программы	Что проверяется данной операцией?
1	При вводе пользователем неподходящего пункта меню.	Программа выводит “Возникла ошибка при вводе. Повторите попытку...”	Устойчивость программы к вводу неверной команды меню
2	Заполнение элементов матрицы не целыми числами.	Программа выводит “Возникла ошибка при вводе. Повторите попытку...”	Устойчивость программы к вводу неверной команды внутри пунктов меню
3	Пользователем введен несуществующий или пустой файл	Программа выводит “Допущена ошибка при открытии файла. Повторите попытку...”	Устойчивость к открытию несуществующего или пустого файла
4	<p>Укажите пункт меню (1-3): 1</p> <p>Размерность матрицы через пробел: 4 4</p> <p>Количество ненулевых элементов: 2</p> <p>Номер строки: 0</p> <p>Номер столбца: 0</p> <p>Укажите значение: 1</p> <p>Номер строки: 1</p> <p>Номер столбца: 0</p> <p>Укажите значение: 40</p>	<p>Укажите пункт меню (0-5): 5</p> <pre> 1 0 0 0 40 0 0 0 0 0 0 0 0 0 0 0 </pre>	Корректный вывод матрицы в стандартной форме

5	<p>Укажите пункт меню (1-3): 1</p> <p>Размерность матрицы через пробел: 4 4</p> <p>Количество ненулевых элементов: 2</p> <p>Номер строки: 0</p> <p>Номер столбца: 0</p> <p>Укажите значение: 1</p> <p>Номер строки: 1</p> <p>Номер столбца: 0</p> <p>Укажите значение: 40</p>	<p>Укажите пункт меню (0-5): 4</p> <p>A: 1 40</p> <p>IA: 0 1</p> <p>JA: 0 2 2 2 2</p>	Вывод матрицы в разреженной форме
6	<p>Укажите пункт меню (1-3): 3</p> <p>Введите размер матрицы и процент заполнения:</p> <p>Высота = 5</p> <p>Ширина = 5</p> <p>Заполненность = 50</p> <p>Матрица генерируется...</p> <p>Меню операций:</p> <p>1 - Умножение вектора-строки на матрицу, хранящихся в разреженной форме, с получением результата в той же форме</p> <p>2 - Умножение стандартным алгоритмом работы с матрицами</p> <p>3 - Сравнение времени выполнения операций, объема памяти при использовании этих алгоритмов при различной доли заполненности матриц</p> <p>4 - Вывод в разреженной форме</p>	<pre> 262 17 0 0 372 0 0 0 0 0 635 101 0 829 624 0 931 352 0 428 0 535 0 0 168 </pre>	Генерация матрицы

	<p>5 - Вывод в простой форме 0 - Завершение программы</p> <p>Укажите пункт меню (0-5): 5</p>		
7	<p>Укажите пункт меню (1-3): 3</p> <p>Введите размер матрицы и процент заполнения:</p> <p>Высота = 10000</p> <p>Ширина = 10000</p> <p>Заполненность = 60</p> <p>Матрица генерируется...</p> <p>Меню операций:</p> <p>1 - Умножение вектора-строки на матрицу, хранящихся в разреженной форме, с получением результата в той же форме</p> <p>2 - Умножение стандартным алгоритмом работы с матрицами</p> <p>3 - Сравнение времени выполнения операций, объема памяти при использовании этих алгоритмов при различной доли заполненности матриц</p> <p>4 - Вывод в разреженной форме</p> <p>5 - Вывод в простой форме</p> <p>0 - Завершение программы</p> <p>Укажите пункт меню (0-5): 6</p>	<p>Время умножения матрицы на вектор в стандартном виде: 897260242 тиков</p> <p>Память умножения матрицы на вектор в стандартном виде: 400000000 байт</p> <p>Время умножения матрицы на вектор в разреженном виде: 995532466 тиков</p> <p>Память умножения матрицы на вектор в разреженном виде: 136442616 байт</p>	Вывод сравнительной информации

7. Выводы по проделанной работе

Табл. 3

Сравнение работы алгоритмов					
Размерность	Разреженность, %	Стандартная		Разреженная	
		Время, сек	Память, байт	Время, сек	Память, байт
200 x 200	0	0.000122	160000	0.000115	800
	10	0.000105		0.000127	288800
	20	0.000103		0.000171	256800
	30	0.000105		0.000199	224800
	40	0.000104		0.000225	192800
	50	0.000103		0.000246	160800
	60	0.000103		0.000215	128800
	70	0.000104		0.000174	96800
	75	0.000103		0.000152	80800
	80	0.000103		0.000128	64800
	85	0.000102		0.000109	48800
	90	0.000102		0.000090	32800
	95	0.000101		0.000074	16800
	100	0.000100		0.000039	4000
300 x 300	0	0.000323	360000	0.000250	721200
	10	0.000321		0.000315	649200
	20	0.000312		0.000368	577200
	30	0.000325		0.000426	505200
	40	0.000322		0.000490	433200
	50	0.000321		0.000528	361200
	60	0.000310		0.000479	289200
	70	0.000332		0.000379	217200
	75	0.000330		0.000328	181200

	80	0.000323		0.000287	145200
	85	0.000320		0.000236	109200
	90	0.000333		0.000225	73200
	95	0.000325		0.000155	37200
	100	0.000353		0.000101	8400
500 x 500	0	0.001128	1000000	0.000710	2002000
	10	0.001217		0.000878	1802000
	20	0.001193		0.001021	1602000
	30	0.001172		0.001186	1402000
	40	0.001208		0.001373	1202000
	50	0.001143		0.001433	1002000
	60	0.001332		0.001336	802000
	70	0.001157		0.001047	602000
	75	0.001161		0.000905	502000
	80	0.001099		0.000765	402000
	85	0.001124		0.000648	302000
	90	0.001098		0.000540	202000
	95	0.001149		0.000434	102000
	100	0.001186		0.000304	22000
800 x 800	0	0.003009	2560000	0.001761	5123200
	10	5123200		0.002365	4611200
	20	0.003346		0.002590	4099200
	30	0.002938		0.002976	3587200
	40	0.002917		0.003470	3075200
	50	0.003019		0.003683	2563200
	60	0.003015		0.003352	2051200
	70	0.003296		0.002680	1539200
	75	0.002972		0.002268	1283200
	80	0.002907		0.001933	1027200

	85	0.003083		0.001659	771200
	90	0.002933		0.001383	515200
	95	0.002806		0.001094	259200
	100	0.002791		0.000809	54400
1000 x 1000	0	0.006180	4000000	0.002788	8004000
	10	0.005971		0.003452	7204000
	20	0.005998		0.004106	6404000
	30	0.005476		0.004677	5604000
	40	0.006451		0.005377	4804000
	50	0.006323		0.005837	4004000
	60	0.005122		0.005221	3204000
	70	0.008012		0.004145	2404000
	75	0.008605		0.003557	2004000
	80	0.005014		0.003074	1604000
	85	0.005687		0.002665	1204000
	90	0.006297		0.002151	804000
	95	0.006411		0.001771	404000
	100	0.005995		0.001292	84000
2000 x 2000	0	0.054001	16000000	0.010659	32008000
	10	0.054809		0.013630	28808000
	20	0.053765		0.016127	25608000
	30	0.054805		0.018557	22408000
	40	0.053605		0.021123	19208000
	50	0.054109		0.022752	16008000
	60	0.054169		0.020667	12808000
	70	0.053446		0.016494	9608000

	75	0.054210		0.013957	8008000
	80	0.054166		0.011969	6408000
	85	0.053361		0.010478	4808000
	90	0.052999		0.008408	3208000
	95	0.053126		0.007096	1608000
	100	0.054918		0.005432	328000
3000 x 3000	0	0.125130	36000000	0.024175	72012000
	10	0.126471		0.030158	64812000
	20	0.115916		0.037282	57612000
	30	0.118292		0.041982	50412000
	40	0.121750		0.048675	43212000
	50	0.121391		0.051986	36012000
	60	0.133597		0.046959	28812000
	70	0.126072		0.036034	21612000
	75	0.133524		0.031354	18012000
	80	0.119878		0.026955	14412000
	85	0.128467		0.023030	10812000
	90	0.127573		0.019204	7212000
	95	0.115838		0.015269	3612000
	100	0.123461		0.012088	732000

Полученные результаты позволяют сделать следующие выводы:

- 1) Разреженный вид становится эффективней по памяти при разреженности более чем 50-60%;
- 2) Разреженный вид становится эффективней по времени при разреженности более чем 75-80%;

- 3) При разреженности меньше 40% разреженный вид гораздо менее эффективный, нежели классический;
- 4) Чем больше матрица, тем сильнее проявляется эффективность разреженного метода хранения (при соответствующих разреженностях).

Таким образом:

- 1) Программист не должен использовать разреженный способ хранения матриц при разреженности меньше 40%.
- 2) Программист должен использовать разреженный способ хранения матриц при разреженности больше 70%.
- 3) В остальных случаях выбор метода решения остаётся за программистом и зависит от доступных ресурсов и поставленных целей.

Использование разреженной матрицы оправдано при большой размерности и небольшом заполнении. В таком случае можно получить значительный выигрыш по памяти и по времени. Однако в этом случае значительно усложняется алгоритм обработки матриц.

Время выполнения стандартного алгоритма почти линейно зависит от размерности матрицы. Этот алгоритм эффективен при высоком заполнении матрицы. Однако при заполнении матрицы менее 20-15% разреженный алгоритм позволяет добиться более высокой скорости работы при использовании меньшего количества памяти.

Сложность разреженного алгоритма связана с невозможностью доступа к элементам по индексам.

При заполнении матрицы более, чем на 15-20%, и размерности менее 50*50, безусловно, целесообразнее использовать стандартные способы обработки и хранения матриц.

8. Контрольные вопросы

1. Что такое разреженная матрица, какие схемы хранения таких матриц Вы знаете?

Разреженная матрица – матрица с преимущественно нулевыми элементами.

Число ненулевых элементов в матрице порядка n может выражаться как n^{1+g} , где $g < 1$. Значения g лежат в интервале $0.2 \dots 0.5$.

Существуют различные методы хранения элементов матрицы в памяти.

Например, линейный связный список, т.е. последовательность ячеек, связанных в определенном порядке. Каждая ячейка списка содержит элемент списка и указатель на положение следующей ячейки.

Можно хранить матрицу, используя кольцевой связный список, двунаправленные стеки и очереди.

Существует диагональная схема хранения симметричных матриц, а также связные схемы разреженного хранения.

Связная схема хранения матриц, предложенная Кнудом, предлагает хранить в массиве (например, в AN) в произвольном порядке сами элементы, индексы строк и столбцов соответствующих элементов (например, в массивах I и J), номер (из массива AN) следующего ненулевого элемента, расположенного в матрице по строке

(NR) и по столбцу (NC), а также номера элементов, с которых начинается строка (указатели для входа в строку – JR) и номера элементов, с которых начинается столбец (указатели для входа в столбец – JC).

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

Для хранения обычной матрицы необходимо: $N * M * \text{sizeof}(\text{elem})$ памяти.

Память под разреженную матрицу выделяется в зависимости от схемы хранения.

Память выделяется по мере наполнения ненулевыми элементами.

3. Каков принцип обработки разреженной матрицы?

Обработка разреженной матрицы предполагает работу только с ненулевыми элементами (таким образом, количество операций пропорционально количеству ненулевых элементов).

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

Нам следует учитывать разреженность матрицы, если из этого можно извлечь выгоду за счёт игнорирования нулевых элементов.

Нужно заметить, что происходит падение эффективности по времени при достижении определенного процента заполненности матрицы ненулевыми элементами.