

	<p>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 (ИУ) _____

КАФЕДРА _____ Программное обеспечение ЭВМ и информационные технологии (ИУ7) _____

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3 **«ОБРАБОТКА РАЗРЕЖЕННЫХ МАТРИЦ»**

Студент, группа

Кононенко С., ИУ7-33Б

2019 г.

Описание условия задачи

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

- вектор **A** содержит значения ненулевых элементов;
 - вектор **IA** содержит номера строк для элементов вектора **A**;
 - связный список **JA**, в элементе **Nk** которого находится номер компонент в **A** и **IA**, с которых начинается описание столбца **Nk** матрицы **A**.
1. Смоделировать операцию сложения двух матриц, хранящихся в этой форме, с получением результата в той же форме.
 2. Произвести операцию сложения, 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.

Техническое задание

Входные данные:

1. **Целое число, представляющее собой номер команды:** целое число в диапазоне от 0 до 6.
2. **Командно-зависимые данные:**
 - количество строк/столбцов матрицы, количество элементов матрицы, элементы матрицы в формате “индекс строки - индекс столбца – значение элемента”;
 - количество строк/столбцов матрицы, процент ненулевых элементов матрицы.

Выходные данные:

1. Исходные и результирующая матрицы в стандартном виде или разреженном столбцовом виде.
2. Количественная характеристика сравнения вариантов сложения матриц.

Функция программы: программа выполняет ряд функций, указанных при её первом запуске. Она позволяет:

1. Ввести матрицы вручную (в формате Matrix Market).
2. Сгенерировать матрицы случайно (по проценту заполненности).
3. Сложить матрицы, используя стандартный способ хранения.
4. Сложить матрицы, используя разреженный столбцовый вид хранения.
5. Вывести исходные и результирующую матрицы в стандартном виде.
6. Вывести исходные и результирующую матрицы в разреженном столбцовом виде.

Обращение к программе: запускается из терминала.

Аварийные ситуации:

1. Некорректный ввод номера команды.
На входе: число, большее чем 6 или меньшее, чем 0.
На выходе: сообщение «Введена недопустимая команда! Повторите попытку.»
2. Некорректный ввод количества строк или столбцов матрицы.
На входе: неположительное целое число или буква.
На выходе: сообщение «Введено недопустимое значение! Повторите попытку.»
3. Некорректный ввод индекса строки или столбца матрицы.
На входе: число, выходящее за границы [0; количество_столбцов(строк)), или буква.
На выходе: сообщение «Введено недопустимое значение! Повторите попытку.»
4. Некорректный ввод элемента матрицы.
На входе: число, выходящее за границы условия, или буква.
На выходе: сообщение «Введено недопустимое значение! Повторите попытку.»

Структуры данных

За стандартное хранение матрицы отвечает именованная структура **matrix_t**, описанная как:

```
typedef struct
{
    type_t **matrix;
    int rows;
    int columns;
} matrix_t;
```

Поля структуры:

- **type_t **matrix** – массив указателей на строки матрицы (**type_t - int**);
- **int rows** – количество строк матрицы;
- **int columns** – количество столбцов матрицы.

За хранение матрицы в разреженном столбцовом виде отвечает именованная структура **sparse_t**, описанная как:

```
typedef struct
{
    type_t *elems;
    int *row_entry;
    int *col_entry;
    int elems_amount;
    int cols_amount;
} matrix_t;
```

Поля структуры:

- **type_t *elems** – массив элементов матрицы, заполняемый проходом по столбцам;
- **int *row_entry** – массив, каждый элемент которого равен номеру строки соответствующего элемента из **elems**;
- **int *col_entry** – массив, каждый элемент которого указывает на индекс элемента из **elems**, с которого начинается описание столбца;
- **int elems_amount** – количество элементов матрицы;
- **int cols_amount** – количество столбцов матрицы.

Алгоритм

1. Пользователь вводит номер команды из меню.
2. Пока пользователь не введет 0 (выход из программы), ему будет предложено выполнять действия с матрицами.
3. При вводе (или генерации) матрицы, матрица сразу хранится двумя способами хранения (стандартном и разреженном столбцовом).
4. В случае выбора стандартного сложения, матрицы складываются в стандартном виде “элемент к элементу”.
5. В случае выбора разреженного столбцового сложения, действия производятся непосредственно над разреженными столбцовыми матрицами. Сравнивается каждый столбец, при этом каждый из них сначала рассматривается как массив построчных вхождений. Если в массивах нет одинаковых вхождений, то в итоговую матрицу элементы записываются по порядку, в порядке возрастания по двум массивам. Если же одинаковые вхождения есть, то соответствующие элементы складываются, после все элементы вновь записываются в порядке возрастания индексов вхождения.

Тесты

	Тест	Пользовательский ввод	Результат
1	Некорректный ввод команды	7	Введена недопустимая команда! Повторите попытку.
2	Некорректный ввод количества строк	0	Введено недопустимое значение! Повторите попытку
3	Некорректный ввод количества строк	A	Введено недопустимое значение! Повторите попытку
4	Некорректный ввод количества столбцов	-2	Введено недопустимое значение! Повторите попытку
5	Некорректный ввод количества столбцов	B	Введено недопустимое значение! Повторите попытку
6	Некорректный ввод индекса строки	При матрице 4x4: 4	Введено недопустимое значение! Повторите попытку
7	Некорректный ввод индекса строки	A	Введено недопустимое значение! Повторите попытку
8	Некорректный ввод индекса столбца	При матрице 3x3: 5	Введено недопустимое значение! Повторите попытку
9	Некорректный ввод индекса столбца	C	Введено недопустимое значение! Повторите попытку
10	Некорректный ввод элемента матрицы	D	Введено недопустимое значение! Повторите попытку
11	Некорректный ввод элемента матрицы	2.1	Введено недопустимое значение! Повторите попытку

Оценка эффективности

Измерения эффективности способов сложения будут производиться в единицах измерения – тактах процессоров. Для измерения была специально написана ассемблерная функция, поэтому погрешность измерений минимальна. При записи результатов использовалось среднее количество тактов, полученное по результатам 10 измерений.

Время сложения:

5% заполнения

Размеры	Разреженная матрица	Обычная матрица
10x10	1060	11900
100x100	33800	1549820
200x200	119080	6947080

10% заполнения

Размеры	Разреженная матрица	Обычная матрица
10x10	1160	8860
100x100	66920	579780
200x200	227680	1590860

20% заполнения

Размеры	Разреженная матрица	Обычная матрица
10x10	1540	7620
100x100	310940	388600
200x200	437380	3946040

30% заполнения

Размеры	Разреженная матрица	Обычная матрица
10x10	7200	20120
100x100	432400	1293560
200x200	648280	1544520

40% заполнения

Размеры	Разреженная матрица	Обычная матрица
10x10	11300	5820
100x100	11847640	388600
200x200	46828840	1608580

50% заполнения

Размеры	Разреженная матрица	Обычная матрица
10x10	25680	8020
100x100	6765340	1816860
200x200	52933140	1568680

100% заполнения

Размеры	Разреженная матрица	Обычная матрица
10x10	40380	6180
100x100	31208440	1293960
200x200	221170380	1763540

Объём занимаемой памяти (в байтах):**5% заполнения**

Размеры	Разреженная матрица	Обычная матрица
10x10	168	648
100x100	12408	6408
200x200	37080	12808

10% заполнения

Размеры	Разреженная матрица	Обычная матрица
10x10	248	648
100x100	20840	6408
200x200	68936	12808

20% заполнения

Размеры	Разреженная матрица	Обычная матрица
10x10	408	648
100x100	37112	6408
200x200	133368	12808

30% заполнения

Размеры	Разреженная матрица	Обычная матрица
10x10	568	648
100x100	53304	6408
200x200	196888	12808

40% заполнения

Размеры	Разреженная матрица	Обычная матрица
10x10	728	648
100x100	69576	6408
200x200	261400	12808

50% заполнения

Размеры	Разреженная матрица	Обычная матрица
10x10	888	648
100x100	84936	6408
200x200	326104	12808

100% заполнения

Размеры	Разреженная матрица	Обычная матрица
10x10	1688	648
100x100	1698782	6408
200x200	641608	12808

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, столбцовый 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 $Z * K$ 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 40%). Стоит отметить, что если не так важна память, занимаемая матрицами, но важно время, то в случае сложения лучше так же воспользоваться стандартными алгоритмами сложения матриц.

Вывод

Использование алгоритмов хранения и обработки разреженных матриц в случае сложения выгодно при малом количестве ненулевых элементов, примерно до 40% заполненности матриц. Хранение матриц в разреженном столбцовом формате проигрывает по памяти практически во всех случаях, так как структура для хранения разреженных матриц довольно нагружена целочисленными полями для хранения различных компонент матриц. Помимо занимаемой памяти, разреженные матрицы гораздо дольше обрабатываются при заполненности матриц от 40%, так как требуется много времени для составления портрета матрицы.