

	<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 (ИУ) _____

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

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

Студент, группа

Буланый К., ИУ7-36Б

2020 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. **Данные, зависящие от команды:**
 - количество строк/столбцов матрицы, количество элементов матрицы, элементы матрицы в формате “**строка - столбец – значение**”;
 - количество строк/столбцов матрицы, процент ненулевых элементов матрицы.

Выходные данные:

1. Исходные и результирующая матрицы в стандартном виде или разреженном столбцовом виде.
2. Время сложения матриц.

Функция программы:

1. Ввести матрицы вручную.
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 «Invalid command.»
2. Некорректный ввод количества строк или столбцов матрицы.
На входе: неположительное целое число или буква.
На выходе: сообщение «Invalid value.»
3. Некорректный ввод индекса строки или столбца матрицы.
На входе: число, выходящее за границы, или буква.
На выходе: сообщение «Invalid value.»
4. Некорректный ввод элемента матрицы.
На входе: число, выходящее за границы условия, или буква.
На выходе: сообщение «Invalid value.»

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.
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	30	Invalid command.
2	Некорректный ввод	0	Invalid value.

	количества строк		
3	Некорректный ввод количества строк	M	Invalid value.
4	Некорректный ввод количества столбцов	-3	Invalid value.
5	Некорректный ввод количества столбцов	P	Invalid value.
6	Некорректный ввод индекса строки	При матрице 2x2: 4	Invalid value.
7	Некорректный ввод индекса строки	S	Invalid value.
8	Некорректный ввод индекса столбца	При матрице 2x2: 5	Invalid value.
9	Некорректный ввод индекса столбца	K	Invalid value.
10	Некорректный ввод элемента матрицы	Q	Invalid value.
11	Некорректный ввод элемента матрицы	3.14	Invalid value.

Оценка эффективности

Время сложения:

5% заполнения

Размеры	Разреженная матрица	Обычная матрица
10x10	1060	11900
50x50	33800	1549820
100x100	119080	6947080

10% заполнения

Размеры	Разреженная матрица	Обычная матрица
10x10	1160	8860

50x50	66920	579780
100x100	227680	1590860

20% заполнения

Размеры	Разреженная матрица	Обычная матрица
10x10	1540	7620
50x50	310940	388600
100x100	437380	3946040

30% заполнения

Размеры	Разреженная матрица	Обычная матрица
10x10	7200	20120
50x50	432400	1293560
100x100	648280	1544520

40% заполнения

Размеры	Разреженная матрица	Обычная матрица
10x10	11300	5820
50x50	11847640	388600
100x100	46828840	1608580

50% заполнения

Размеры	Разреженная матрица	Обычная матрица
10x10	25680	8020
50x50	6765340	1816860
100x100	52933140	1568680

100% заполнения

Размеры	Разреженная матрица	Обычная матрица
10x10	40380	6180
50x50	31208440	1293960
100x100	221170380	1763540

Объём занимаемой памяти (в байтах):

5% заполнения

Размеры	Разреженная матрица	Обычная матрица
---------	---------------------	-----------------

10x10	112	496
50x50	4432	40816
100x100	16832	161616

10% заполнения

Размеры	Разреженная матрица	Обычная матрица
10x10	152	496
50x50	8832	40816
100x100	32832	161616

20% заполнения

Размеры	Разреженная матрица	Обычная матрица
10x10	232	496
50x50	16432	40816
100x100	66456	161616

30% заполнения

Размеры	Разреженная матрица	Обычная матрица
10x10	312	496
50x50	24432	40816
100x100	98336	161616

40% заполнения

Размеры	Разреженная матрица	Обычная матрица
10x10	392	496
50x50	32432	40816
100x100	130664	161616

45% заполнения

Размеры	Разреженная матрица	Обычная матрица
10x10	432	496
50x50	37360	40816
100x100	146304	161616

50% заполнения

Размеры	Разреженная матрица	Обычная матрица
10x10	472	496
50x50	43328	40816

100x100	162584	161616
---------	--------	--------

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 $I * J$ ячеек памяти, где I – строки, а J – столбцы. Для разреженной матрицы – зависит от способа. В случае разреженного столбцового формата, требуется $2 * K + J + 1$ ячеек памяти, где K – количество ненулевых элементов.

3. Каков принцип обработки разреженной матрицы?

Алгоритмы обработки разреженных матриц предусматривают действие только с ненулевыми элементами, 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%). Если не важна память, занимаемая матрицами, но важно время, то в случае сложения лучше так же воспользоваться стандартными алгоритмами сложения матриц.

Вывод

Использование разреженных матриц выгодно при малом количестве ненулевых элементов, до ~40 % заполненности. Хранение в разреженном столбцовом формате почти всегда менее выгодно по памяти, так как структура нагружена полями для хранения различных параметров матрицы. Кроме того, разреженные матрицы гораздо дольше обрабатываются при сильной заполненности, так как требуется много времени для составления матрицы.