

	<p align="center"> Министерство науки и высшего образования 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 «ОБРАБОТКА РАЗРЕЖЕННЫХ МАТРИЦ»

Студент _____ Хамзина Регина Ренатовна _____
фамилия, имя, отчество

Студент, группа
ИУ7-33Б

Хамзина Р.Р.,

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 N_k которого находится номер компонент и в A и IA , с которых начинается описание столбца N_k 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 — номер команды меню, в диапазоне от 0 до 10.
2. При вводе матриц:
 - целое число — количество строк матрицы $[1;1000]$;
 - целое число — количество столбцов матрицы $[1;1000]$;
 - целое число — количество ненулевых элементов матрицы $[1;1000]$;
 - индекс элемента — номер строки $[1;1000]$;
 - индекс элемента — номер столбца $[1;1000]$;
 - целые числа — элементы матрицы $[-\infty;+\infty)$.

Выходные данные:

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

Функция программы:

Сложение матриц при классическом и разреженном столбцовом хранении.

Обращение к программе:

Программа запускается из терминала командой «./app.exe» в директории с программой.

Аварийные ситуации

1. Некорректный ввод пункта меню.

На входе: число, большее чем 10 или меньшее, чем 0.

На выходе: сообщение «Команда введена неверно»

2. Некорректный ввод количества строк или столбцов матрицы.

На входе: неположительное целое число или буква.

На выходе: сообщение «Число строк/столбцов матрицы введено неверно.»

3. Некорректный ввод индекса строки или столбца матрицы.

На входе: число, выходящее за границы интервала [0; количество_столбцов(строк)), или буква.

На выходе: сообщение «Элемент введен неправильно»

4. Некорректный ввод элемента матрицы.

На входе: буква.

На выходе: сообщение «Элемент введен неправильно»

Внутренняя структура данных

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

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

Поля структуры:

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

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

```
typedef struct
{
    int*elements;
    int *rows;
    int *columns;
```

```
    int count_non_zero;  
} sparse_matrix;
```

Поля структуры:

- *int *elements* - массив ненулевых элементов матрицы;
- *int *rows*; – массив, каждый элемент которого равен номеру строки соответствующего элемента из первого массива;
- *int *columns* – массив, каждый элемент которого указывает на индекс элемента из первого массива, с которого начинается описание столбца;
- *int count_non_zero*; – количество ненулевых элементов матрицы;

Алгоритм

1. Пользователь вводит пункт 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 классическим способом “элемент к элементу”.
5. В случае выбора разреженного столбцового сложения, действия производятся непосредственно над разреженными столбцовыми матрицами. Алгоритм разделяется на две части — символическую и численную. В символической части заполняется массив со значениями строк результирующей матрицы — множество значений из элементов массивов строк разреженной матрицы A и разреженной матрицы B для каждого столбца. В численной части — заполняется массив ненулевых значений результирующей матрицы — сначала записываются значения элементов разреженной матрицы A, к ним добавляются значения элементов разреженной матрицы B.

Функции программы

- 1) Ввести матрицу вручную

```
int input_matrix(matrix_t *matrix, sparse_matrix *sparse)
```

- 2) Автоматически заполнить матрицу

```
void random_matrix(matrix_t *const matrix, const int size)
```

- 3) Сложить матрицы, используя классический способ хранения

```
int classic_add(matrix_t matrix_A, matrix_t matrix_B, matrix_t *result, uint64_t *time)
```

- 4) Сложить матрицы, используя разреженный столбцовый способ хранения

```
int sparse_sum(sparse_matrix sparse_a, sparse_matrix sparse_b, sparse_matrix *result, const int column, const int row, uint64_t *time)
```

- 5) Вывод матрицы в классическом виде

int print_matrix(matrix_t matrix);

6) Вывод матрицы в разреженном столбцовом виде

int print_sparse(sparse_matrix matrix, const int count)

Тесты

	Тест	Ввод	Вывод
1	Неверный пункт меню	12	Команда введена неверно
2	Неверный ввод строк	-1	Число строк матрицы введено неверно
3	Неверный ввод столбцов	0	Число столбцов матрицы введено неверно
4	Некорректный ввод индекса строки	При матрице 4x4: 5	Элемент введен неправильно
5	Некорректный ввод элемента	Ф	Элемент введен неправильно
6	Обычный тест	3 3 3 1 1 1 2 2 2 3 3 3 3 3 3 1 1 3 2 2 2 3 3 1	4 0 0 0 4 0 0 0 4 4 4 4 1 2 3 1 2 3

Оценка эффективности

Время сложения (тики):

5% заполнения

Размер	Обычная матрица	Разреженная матрица
10x10	2184	936
100x100	95784	23452
200x200	1421108	126932

10% заполнения

Размер	Обычная матрица	Разреженная матрица
10x10	5434	2314
100x100	245986	52260
200x200	1628094	121342

20% заполнения

Размер	Обычная матрица	Разреженная матрица
10x10	4758	3744
100x100	322816	88790
200x200	1383798	207974

30% заполнения

Размер	Обычная матрица	Разреженная матрица
10x10	6188	5304
100x100	237796	121446
200x200	1180920	273130

40% заполнения

Размер	Обычная матрица	Разреженная матрица
10x10	4472	5148
100x100	123266	264940
200x200	1401738	282152

50% заполнения

Размер	Обычная матрица	Разреженная матрица
10x10	4810	5096
100x100	147134	412282
200x200	296894	1166438

100% заполнения

Размер	Обычная матрица	Разреженная матрица
10x10	4082	7878
100x100	165880	263380
200x200	497224	1383824

Объем занимаемой памяти (байты):

5% заполнения

Размер	Обычная матрица	Разреженная матрица
10x10	496	136
100x100	40816	11764
200x200	161616	11812

10% заполнения

Размер	Обычная матрица	Разреженная матрица
10x10	496	148

100x100	40816	8828
200x200	161616	32828

20% заполнения

Размер	Обычная матрица	Разреженная матрица
10x10	496	228
100x100	40816	16428
200x200	161616	66454

30% заполнения

Размер	Обычная матрица	Разреженная матрица
10x10	496	308
100x100	40816	22428
200x200	161616	98332

40% заполнения

Размер	Обычная матрица	Разреженная матрица
10x10	496	388
100x100	40816	32428
200x200	161616	130660

50% заполнения

Размер	Обычная матрица	Разреженная матрица
10x10	496	468
100x100	40816	43324
200x200	161616	162580

100% заполнения

Размер	Обычная матрица	Разреженная матрица
10x10	496	468
100x100	40816	43324
200x200	161616	162580

Ответы на контрольные вопросы

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

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 столбцового формата, требуется $2 * K + T$ ячеек памяти, где K – количество ненулевых элементов, T — число столбцов.

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.

Выводы

Хранение матриц в разреженном столбцовом формате нагружено целочисленными полями для хранения различных компонент матриц, при этом разреженные матрицы гораздо дольше обрабатываются при заполненности матриц от 40%, так как требуется много времени для составления портрета матрицы.