



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)

ФАКУЛЬТЕТ «Информатика и системы управления»
КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Отчёт по лабораторной работе №3 по курсу «Типы и структуры данных»

Студент: Шавиш Тарек
Группа: ИУ7и-31Б
Преподаватель: Силантьева А. В.

2023 г

Оглавление

Описание условия задачи	1
Описание технического задания	2
Входные данные	3
Выходные данные	4
Функции меню в программе	5
Обращение к программе	6
Аварийные ситуации	7
Описание структур данных	8
Описание функций	9
Описание алгоритма	10
Набор тестов	11
Сравнение алгоритмов сложения матриц	12
Временные замеры	13
Используемая память	14
Вывод	15
Ответы на контрольные вопросы	16

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 ;
- вектор JA , в элементе Nk которого находится номер компонент в A и IA , с которых начинается описание столбца Nk матрицы A .

Задание:

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.

Описание технического задания

Исходные данные

- Целое число — выбор способа заполнения матрицы (с клавиатуры или из файла).
- Целые числа — количество строк, количество столбцов, количество ненулевых элементов, элементы матрицы (если выбран ввод с клавиатуры).
- Файл с матрицей — в случае выбора ввода из файла.
- Целое число — выбор действия, выполняемого с матрицами.

Результат

- Матрица в формате CSC — при выборе вывода результата в этом формате.
- Матрица в стандартном формате — при выборе вывода результата в этом формате.
- Информация о времени выполнения и объеме памяти, занимаемом матрицами в разных форматах — при выборе измерения времени и памяти.

Функциональные возможности программы

Ввод матрицы:

1. Вручную ввести матрицу с клавиатуры.
2. Загрузить матрицу из файла.

Операции с матрицами:

1. Сложить матрицы и вывести результат в формате CSC.
2. Сложить матрицы и вывести результат в формате CSC и стандартном формате.
3. Измерить время выполнения сложения матриц в формате CSC и стандартном формате.
4. Определить объем памяти, занимаемый матрицами и их суммой.

Запуск программы

Для запуска программы используйте команду в терминале:
./app

Ситуации, требующие особого внимания:

1. Неправильный выбор пункта в меню.
2. Ввод несуществующего имени файла.
3. Выбор файла, который не содержит матрицу.

4. Обнаружение повторяющегося элемента в файле с матрицей.
5. Произошла ошибка при выделении памяти.
6. Размеры матриц не соответствуют для выполнения операции сложения.

Описание структур данных

1. Для хранения разреженной матрицы используется структура `sparse_matrix_t`:

```
```\ntypedef struct {\n    size_t nrows;    // Количество строк\n    size_t ncols;    // Количество столбцов\n    size_t nelems;    // Количество ненулевых элементов\n    int *A;          // Массив элементов матрицы\n    size_t *IA;       // Массив номеров строк, где расположены\nсоответствующие элементы\n    size_t *JA;       // Массив, в элементе Nk которого находится\nномер компонента в A и IA, с которого начинается описание\nстолбца Nk матрицы A\n} sparse_matrix_t;\n```\n
```

Поля структуры:

- `nrows`: количество строк;
- `ncols`: количество столбцов;
- `nelems`: количество ненулевых элементов;
- `A`: массив элементов матрицы;

- IA: массив номеров строк, где располагаются соответствующие элементы матрицы;
- JA: массив, в элементе Nk которого находится номер компонента в A и IA, с которого начинается описание столбца Nk матрицы A.

2. Для хранения матрицы в стандартном виде в памяти используется структура `matrix_t`:

```
```c
typedef struct {
    int **data;    // Двумерный массив элементов матрицы
    size_t nrow;   // Количество строк матрицы
    size_t ncol;   // Количество столбцов матрицы
} matrix_t;
```
```

Поля структуры:

- data: двумерный массив элементов матрицы;
- nrow: количество строк матрицы;
- ncol: количество столбцов матрицы.

Описание функций

1. `sparse_matrix_input`: Функция для ввода разреженной матрицы с клавиатуры в координатном формате. Принимает указатель на разреженную матрицу. Возвращает код ошибки.

```
```c
int sparse_matrix_input(sparse_matrix_t *sparse_matrix);
```
```

2. ``sparse_matrix_alloc``: Функция для выделения памяти под разреженную матрицу в разреженном столбцовом формате. Принимает указатель на разреженную матрицу. Возвращает код ошибки.

```
```c
int sparse_matrix_alloc(sparse_matrix_t *sparse_matrix);
```
```

3. ``sparse_matrix_free``: Функция для освобождения памяти, занимаемой разреженной матрицей. Принимает указатель на разреженную матрицу.

```
```c
void sparse_matrix_free(sparse_matrix_t *sparse_matrix);
```
```

4. ``sparse_matrix_to_matrix``: Функция для преобразования разреженной матрицы в стандартную матрицу. Принимает указатель на разреженную матрицу и возвращает стандартную матрицу. Возвращает код ошибки.

```
```c
int sparse_matrix_to_matrix(matrix_t *matrix, sparse_matrix_t
*sparse_matrix);
```
```

5. ``sparse_matrix_fscan``: Функция для считывания разреженной матрицы из файла. Принимает указатель на разреженную матрицу и файл. Возвращает код ошибки.

```
```c
```

```
int sparse_matrix_fscan(sparse_matrix_t *sparse_matrix, FILE
*f);
```
```

6. `sparse\_matrices\_add`: Функция для сложения двух разреженных матриц. Принимает указатели на результат матрицы, первую и вторую матрицы. Возвращает код ошибки.

```
```c
int sparse_matrices_add(sparse_matrix_t *matrix_res,
sparse_matrix_t *matrix_1, sparse_matrix_t *matrix_2);
```
```

7. `time\_sparse\_matrices\_add`: Функция для измерения времени сложения двух разреженных матриц. Принимает указатель на переменную времени, указатели на результат матрицы, первую и вторую матрицы. Возвращает код ошибки.

```
```c
int time_sparse_matrices_add(microsecond_t *time,
sparse_matrix_t *matrix_res, sparse_matrix_t *matrix_1,
sparse_matrix_t *matrix_2);
```
```

8. `sparse\_matrix\_output`: Функция для вывода разреженной матрицы в разреженном столбцовом формате. Принимает указатель на разреженную матрицу.

```
```c
void sparse_matrix_output(sparse_matrix_t *sparse_matrix);
```
```



9. ``sparse_matrix_size``: Функция для подсчета размера, занимаемого в памяти разреженной матрицей. Принимает указатель на разреженную матрицу. Возвращает размер.

```
```c
size_t sparse_matrix_size(sparse_matrix_t *sparse_matrix);
```
```

10. ``matrix_free``: Функция для освобождения памяти, занимаемой стандартной матрицей. Принимает указатель на стандартную матрицу.

```
```c
void matrix_free(matrix_t *matrix);
```
```

11. ``matrix_output``: Функция для вывода стандартной матрицы в стандартном формате. Принимает указатель на стандартную матрицу.

```
```c
void matrix_output(matrix_t *matrix);
```
```

12. ``matrices_add``: Функция для сложения двух стандартных матриц. Принимает указатели на результат матрицы, первую и вторую матрицы. Возвращает код ошибки.

```
```c
int matrices_add(matrix_t *matrix_res, matrix_t *matrix_1,
matrix_t *matrix_2);
```
```

13. ``matrix_alloc``: Функция для выделения памяти под стандартную матрицу. Принимает указатель на стандартную матрицу. Возвращает код ошибки.

```
```c  
int matrix_alloc(matrix_t *matrix);  
```
```

14. ``time_matrices_add``: Функция для измерения времени сложения двух стандартных матриц. Принимает указатель на переменную времени, указатели на результат матрицы, первую и вторую матрицы. Возвращает код ошибки.

```
```c  
int time_matrices_add(microsecond_t *time, matrix_t  
*matrix_res, matrix_t *matrix_1, matrix_t *matrix_2);  
```
```

15. ``matrix_size``: Функция для подсчета размера, занимаемого в памяти стандартной матрицей. Принимает указатель на стандартную матрицу. Возвращает размер.

## Описание алгоритма

### 1. Выбор входных данных:

- Пользователь выбирает способ ввода первой матрицы: с клавиатуры или из файла.
- Если выбран ввод из файла, пользователь указывает имя файла. Если выбран ввод с клавиатуры, матрица вводится координатно.

### 2. Выбор действия:

- Пользователь выбирает одно из следующих действий:
  - а. Сложение матриц и вывод результата в разреженном столбцовом формате.
  - б. Сложение матриц и вывод результата в разреженном столбцовом формате и в стандартном формате.
  - с. Измерение времени и памяти, занимаемой матрицами в разреженном и стандартном форматах.

### 3. Выполнение действий:

- Если выбрано действие сложения:
  - Разреженные матрицы складываются, и результат выводится в разреженном столбцовом формате.
  - Если выбрано также вывод в стандартном формате, результат выводится в обоих форматах.
  - Если выбрано измерение времени и памяти:
    - Разреженные матрицы складываются как разреженные и как стандартные матрицы.
    - Измеряется время выполнения операции сложения.
  - Подсчитывается объем памяти, занимаемый разреженными и стандартными матрицами.

- Результаты измерений выводятся на экран.

Этот алгоритм позволяет пользователю ввести матрицу, выполнить различные операции с ней и получить результаты сложения, а также измерить время выполнения и использование памяти при различных форматах хранения матриц.

**Тест:**

| № | Название теста                             | Ввод           | Вывод                  |
|---|--------------------------------------------|----------------|------------------------|
| 1 | Некорректный пункт меню                    | 123            | Некорректный выбор.    |
| 2 | Некорректный пункт меню                    | abc            | Некорректный выбор.    |
| 3 | Некорректное имя файла при чтении из файла | 2<br>asas.txt  | Ошибка открытия файла. |
| 4 | Выбор пустого файла                        | 2<br>empty.txt | пуст.                  |

|    |                                              |                                                                               |                                                             |
|----|----------------------------------------------|-------------------------------------------------------------------------------|-------------------------------------------------------------|
| 5  | Выбор файла, не содержащего матрицу          | 2<br>file.txt                                                                 | Файл повреждён или пуст.                                    |
| 6  | Повторение элемента матрицы в файле          | 2<br>file.txt                                                                 | Ошибка: в файле несколько раз указан один и тот же элемент. |
| 7  | Некорректный ввод при вводе с клавиатуры     | 1<br>abc                                                                      | Некорректный ввод.                                          |
| 8  | Ввод элемента, выходящего за границы матрицы | 1<br>2 2 1<br>2 3 1                                                           | Некорректный ввод.                                          |
| 9  | Сложение матриц различной размерности        | 1<br>2 2 1<br>0 0 2<br>2 3 1<br>0 1 10                                        | Ошибка: матрицы имеют различную размерность.                |
| 10 | Ввод с клавиатуры и вывод в формате CSC      | 1<br>5 5 3<br>0 0 5<br>0 1 3<br>1 1 -9<br>1<br>5 5 2<br>1 1 20<br>3 2 25<br>1 | A: 5 3 11 25<br>IA: 0 0 1 3<br>JA: 0 1 3 4 4 4              |

| Плотность<br>(%) | Размерность | Время<br>сложения<br>стандартн<br>-ых матриц<br>, мкс | Время<br>сложения<br>Разреженн<br>-ых Матриц<br>мкс | Отношение<br>времени<br>сложения<br>стандартных<br>матриц к<br>времени<br>сложения<br>разреженных |
|------------------|-------------|-------------------------------------------------------|-----------------------------------------------------|---------------------------------------------------------------------------------------------------|
| 1                | 100x100     | 22                                                    | 11                                                  | 2                                                                                                 |
| 1                | 200x200     | 119                                                   | 51                                                  | 2.33                                                                                              |
| 1                | 500x500     | 828                                                   | 253                                                 | 3.27                                                                                              |
| 5                | 100x100     | 24                                                    | 21                                                  | 1.14                                                                                              |
| 5                | 200x200     | 101                                                   | 109                                                 | 0.93                                                                                              |
| 5                | 500x500     | 776                                                   | 1040                                                | 0.75                                                                                              |
| 10               | 100x100     | 27                                                    | 47                                                  | 0.57                                                                                              |
| 10               | 200x200     | 96                                                    | 233                                                 | 0.04                                                                                              |
| 10               | 500x500     | 777                                                   | 2936                                                | 0.26                                                                                              |
| 12.5             | 100x100     | 26                                                    | 60                                                  | 0.43                                                                                              |
| 12.5             | 200x200     | 94                                                    | 361                                                 | 0.26                                                                                              |
| 12.5             | 500x500     | 752                                                   | 4247                                                | 0.18                                                                                              |
| 15               | 100x100     | 26                                                    | 79                                                  | 0.33                                                                                              |
| 15               | 200x200     | 104                                                   | 471                                                 | 0.22                                                                                              |
| 15               | 500x500     | 829                                                   | 6119                                                | 0.14                                                                                              |
| 20               | 100x100     | 27                                                    | 104                                                 | 0.26                                                                                              |
| 20               | 200x200     | 108                                                   | 726                                                 | 0.15                                                                                              |
| 20               | 500x500     | 842                                                   | 10156                                               | 0.08                                                                                              |

## Сравнение алгоритмов сложения матриц

### Временные замеры

Во всех замерах время сортировки вычисляется как среднее из 100 замеров. Время измеряется в микросекундах.

|    |         |         |         |      |
|----|---------|---------|---------|------|
| 33 | 100x100 | 40000   | 40432   | 0.99 |
| 33 | 500x500 | 1000000 | 994032  | 1.00 |
| 50 | 100x100 | 40000   | 60832   | 0.66 |
| 50 | 500x500 | 1000000 | 1504032 | 0.66 |

## контрольные вопросы

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

Разреженная матрица — матрица с преимущественно нулевыми элементами. При хранении могут использоваться следующие схемы: координатная, связная, разреженная строчная/столбцовая (compressed sparse row/column)

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

При хранении обычной матрицы память выделяется на хранение массива строк/столбцов матрицы. То есть, размер памяти, занимаемой обычной матрицей, будет равен  $nrows * ncols * sizeof(int)$ .

При хранении разреженной матрицы память выделяется на хранение массивов, содержащих элементы и вспомогательные данные, характеризующие положение элемента в матрице. Так, при хранении матрицы в разреженном строчном/столбцовом формате память выделяется под массив элементов, массив соответствующих им индексов столбцов/строк и массив, отмечающий позиции в первых двух массивов, с которых начинается описание очередной строки. То есть, размер памяти будет равен  $sizeof(int) * (2 * nelems + n + 1)$ , где  $nelems$  количество ненулевых элементов,  $n$  — количество строк/столбцов.

### 3. Каков принцип обработки разреженной матрицы?

Принцип обработки разреженной матрицы заключается в том, что обрабатываются только ненулевые элементы, а нулевые элементы матрицы в памяти не хранятся и не обрабатываются.

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

Чем менее разряжена матрица, тем меньше смысла использовать для её обработки алгоритмы для разреженных матриц. В данной реали-



зации алгоритма сложения разреженных матриц обработка стандартными методами становится эффективнее обработки алгоритмами для разреженных матриц по времени примерно при 5–10%, а по памяти примерно при 33%.

## **Вывод**

Алгоритмы обработки разреженных матриц эффективнее, чем стандартные алгоритмы обработки матриц, если матрицы имеют небольшую плотность.

Так, реализованный алгоритм сложения разреженных матриц становится менее эффективным по времени, чем стандартный алгоритм сложения, при 5–10% плотности, а по памяти — при примерно 33% плотности.