



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

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

Студент

Городский Юрий Николаевич

Группа

ИУ7 – 32Б

Оглавление

Условие задачи.....	3
Техническое задание.....	3
Функции программы.....	4
Аварийные ситуации:.....	4
Обращение к программе.....	4
Структура данных.....	4
Тесты.....	5
Замерный эксперимент.....	8
Контрольные вопросы.....	12
Вывод.....	13

Условие задачи

Цель работы: реализация алгоритмов обработки разреженных матриц, сравнение эффективности применения этих алгоритмов со стандартными алгоритмами обработки матриц при различном размере матриц и степени их разреженности.

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 N_k которого находится номер компонент в A и IA , с которых начинается описание столбца N_k матрицы A .
1. Смоделировать операцию умножения вектора-строки хранящегося в форме вектора A и вектора, содержащего номера столбцов этих элементов, и матрицы, хранящейся в указанной форме, с получением результата в форме хранения вектора-строки.
 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.

Техническое задание

Входные данные:

1. **Номер команды:** целое число в диапазоне от 0 до 7
2. **Матрица:** Вводится количество строк, столбцов и ненулевых элементов, после чего вводятся координаты ненулевых элементов и их значение, координаты начинаются с (1,1) (написано в интерфейсе программы).
3. **Вектор-строка:** Вводится количество элементов, ненулевых элементов, после чего вводятся координаты ненулевых элементов и их значение, координаты начинаются с 1 (написано в интерфейсе программы).

Выходные данные:

1. Вектор-строка — результат произведения матрицы и вектора.
2. Файл с результатами временных замеров.

Функции программы

1. Ввести разреженную матрицу.
2. Ввести вектор.
3. Вывести разреженную матрицу.
4. Вывести вектор.
5. Умножить вектор на разреженную матрицу.
6. Умножить вектор на разреженную матрицу в обычном виде.
7. Замерный эксперимент.
8. Выйти.

Аварийные ситуации:

1. Некорректный ввод команды (введено не число или число не находится в диапазоне от 0 до 7): сообщение «Неверная команда».
2. Введено не число при вводе матрицы/вектора-строки: сообщение «Ошибка ввода».
3. Введено неправильное число при вводе матрицы/вектора-строки: сообщение «Ошибка диапазона данных».
4. Не удалось открыть файл при записи замерного эксперимента: сообщение «Не удалось открыть файл».

Обращение к программе

Запуск через терминал (./app.exe).

Структура данных

```
// Разреженная матрицы
typedef struct
{
    size_t n; // Количество строк
    size_t m; // Количество столбцов
    size_t el_num; // Количество ненулевых элементов
    int *values; // Значения ненулевых элементов
    size_t *rows; // Строки ненулевых элементов
    ssize_t *column_coords; // Индексы начала столбцов в values
} sparse_t;
```

```
// Вектор
typedef struct
{
    int *values; // Значения ненулевых элементов
    size_t *columns; // Столбцы ненулевых элементов
    size_t m; // Количество столбцов
    size_t el_num; // Количество ненулевых элементов
} vector_t;
```

Матрица и вектор-строка в обычном виде хранятся в виде двумерных массив с размерами m и n.

```
size_t m_1, n_1;
int **mat_1 = NULL;
```

Тесты

Таблица 1: Негативные тесты

№	Описание	Входные данные	Выходные данные	Ожидаемые выходные данные
1	Некорректные размеры (число)	1 1 0 0	«Ошибка диапазона данных»	«Ошибка диапазона данных»
2	Некорректные размеры (не число)	1 a 1 1	«Ошибка ввода»	«Ошибка ввода»
3	Некорректная команда (число)	8	«Неверная команда»	«Неверная команда»
4	Некорректная команда (не число)	a	«Неверная команда»	«Неверная команда»
5	Перезапись ненулевого элемента	1 2 2 2 1 1 1 1 1 1	«Ошибка: перезапись элемента»	«Ошибка: перезапись элемента»
6	Количество ненулевых	1 2 2 5	«Ошибка диапазона данных»	«Ошибка диапазона данных»

элементов > количества элементов			
--	--	--	--

Таблица 2: Позитивные тесты

№	Описание	Ввод	Вывод	Ожидаемый вывод
1	Ввод матрицы	1 {Корректные данные}	«Операция завершена»	«Операция завершена»
2	Ввод вектора	2 {Корректные данные}	«Операция завершена»	«Операция завершена»
3	Вывод матрицы	3	<Если число столбцов и строк меньше 10> В разреженном виде: {Матрица в разреженном виде по заданию} <Если число столбцов или строк больше 10> В нормальном виде: {Матрица в обычном виде} «Операция завершена»	<Если число столбцов и строк меньше 10> В разреженном виде: {Матрица в разреженном виде по заданию} <Если число столбцов или строк больше 10> В нормальном виде: {Матрица в обычном виде} «Операция завершена»
4	Вывод пустой матрицы	3	«В разреженном виде:» «Пустая матрица» «Операция завершена»	«В разреженном виде:» «Пустая матрица» «Операция завершена»
5	Вывод вектора	4	<Если число столбцов меньше 10> В разреженном виде: {Вектор в разреженном виде по заданию} <Если число столбцов больше 10> В нормальном виде:	<Если число столбцов меньше 10> В разреженном виде: {Вектор в разреженном виде по заданию} <Если число столбцов больше 10> В нормальном виде:

			{Вектор в обычном виде} «Операция завершена»	{Вектор в обычном виде} «Операция завершена»
6	Вывод пустого вектора	4	«В разреженном виде:» «Пустой вектор» «Операция завершена»	«В разреженном виде:» «Пустой вектор» «Операция завершена»
7	Умножение матрицы на вектор	5	<Если число столбцов и строк меньше 10> В разреженном виде: «Результат произведения в разреженном виде» <Если число столбцов и строк больше 10> В нормальном виде: «Результат произведения в разреженном виде» «Операция завершена»	<Если число столбцов и строк меньше 10> В разреженном виде: «Результат произведения в разреженном виде» <Если число столбцов и строк больше 10> В нормальном виде: «Результат произведения в разреженном виде» «Операция завершена»
8	Умножение матрицы на вектор в обычном виде	6	<Если число столбцов и строк меньше 10> В разреженном виде: «Результат произведения в разреженном виде» <Если число столбцов и строк больше 10> В нормальном виде: «Результат произведения в разреженном виде» «Операция завершена»	<Если число столбцов и строк меньше 10> В разреженном виде: «Результат произведения в разреженном виде» <Если число столбцов и строк больше 10> В нормальном виде: «Результат произведения в разреженном виде» «Операция завершена»
9	Замерный эксперимент	7	2 файла с выходными данными замера	2 файла с выходными данными замера

Ввод	Вывод
Введите команду: 1 10 10 5 1 1 1 2 2 3 3 3 4 4 4 5 4 5 6 Введите команду: 3 Введите команду: 2 10 5 1 1 2 2 3 3 5 5 4 4 Введите команду: 4 Введите команду: 3 Введите команду: 5	В нормальном виде: 10 10 1 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 4 0 0 0 0 0 0 0 0 0 5 6 0 Операция завершена ... В нормальном виде: 1 2 3 4 5 0 0 0 0 Операция завершена ... 1 6 12 20 24 0 0 0 0 Операция завершена
Введите команду: 1 20 20 2 1 1 1 3 3 4 Введите команду: 3 Введите команду: 2 20 4 1 1 2 2 3 3 4 4 Введите команду: 4 Введите команду: 5 Введите команду: 6	В разреженном виде: A: 1 4 IA: 0 2 JA: 0 - 1 - - - - - Операция завершена ... В разреженном виде: A: 1 2 3 4 JA: 0 1 2 3 Операция завершена ... A: 1 12 JA: 0 2 Операция завершена ... A: 1 12 JA: 0 2 Операция завершена

Замерный эксперимент

Замеры проводились для матриц размером 500x500 и 1000x1000 для заполнения матрицы в различных диапазонах. Расчеты проводились до $rse < 5\%$.

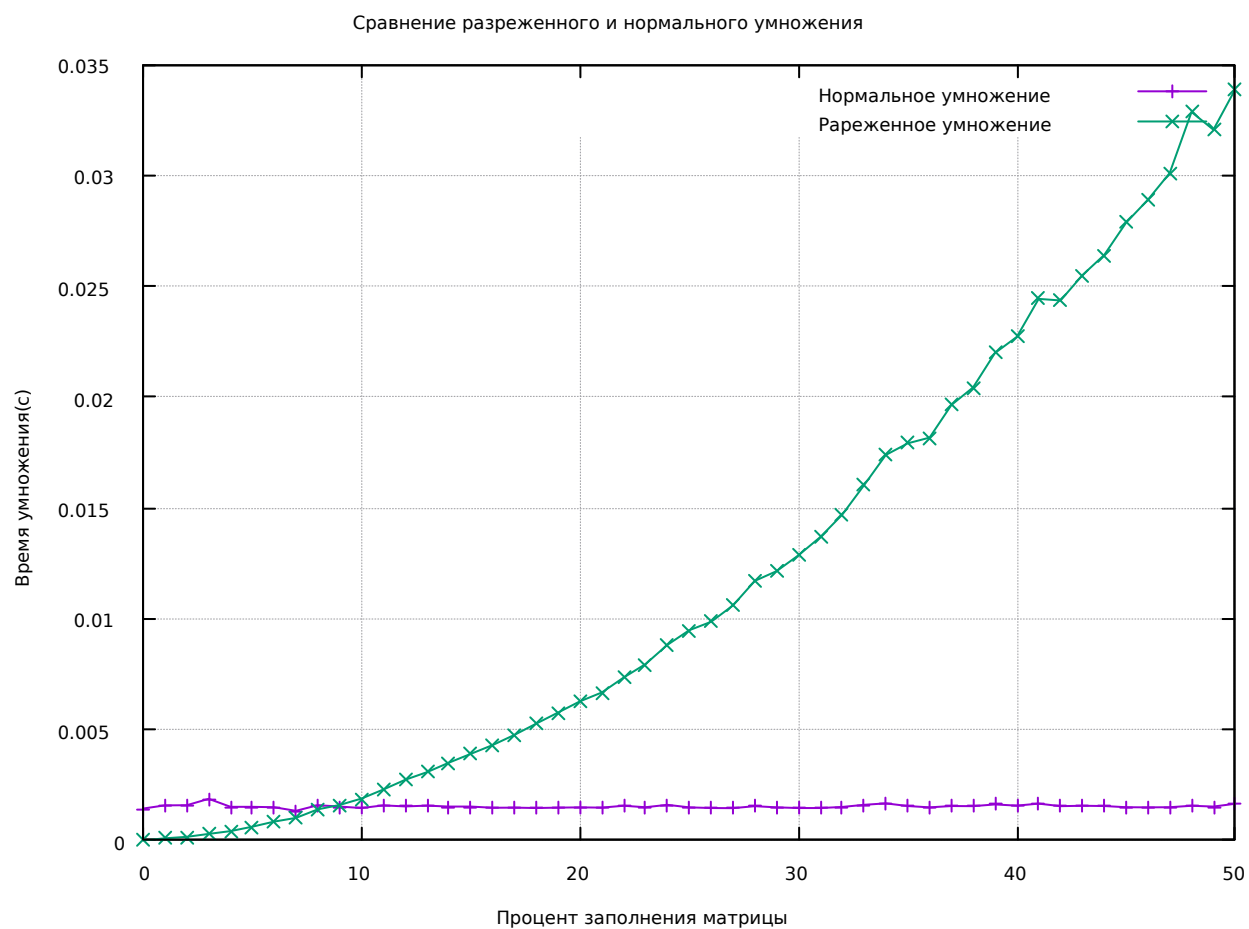


Рисунок 1: 500x500 время умножения

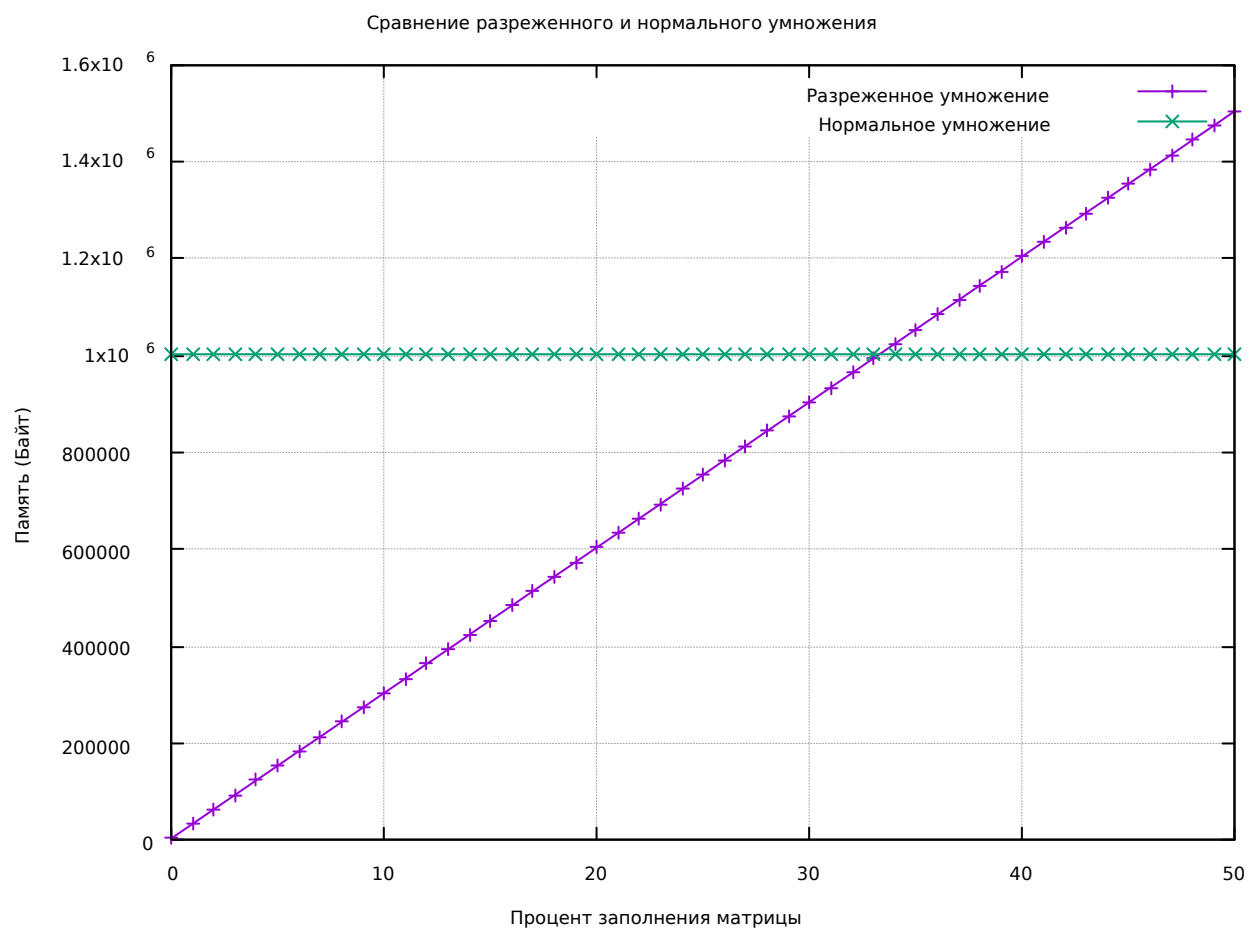
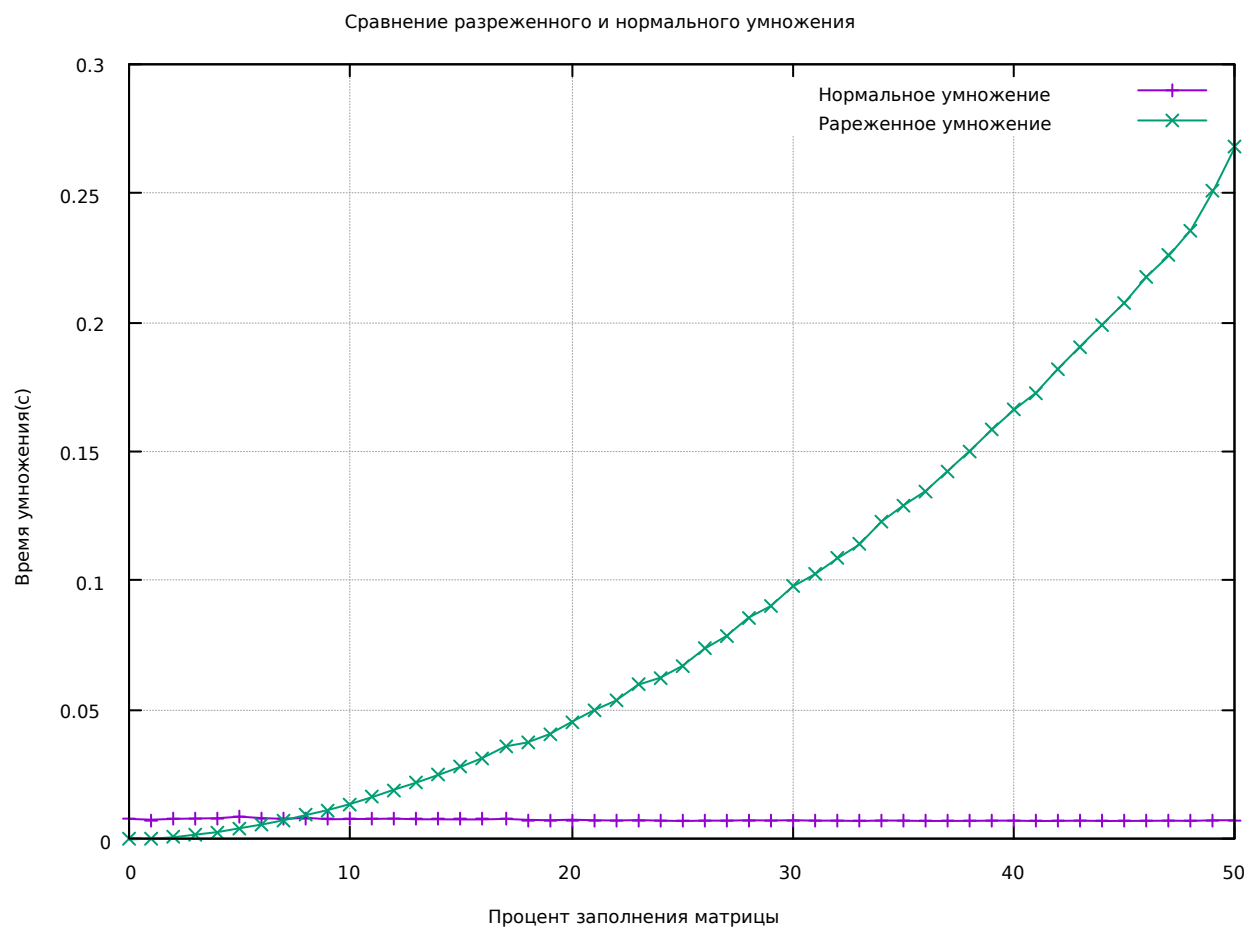


Рисунок 2: 500x500 память



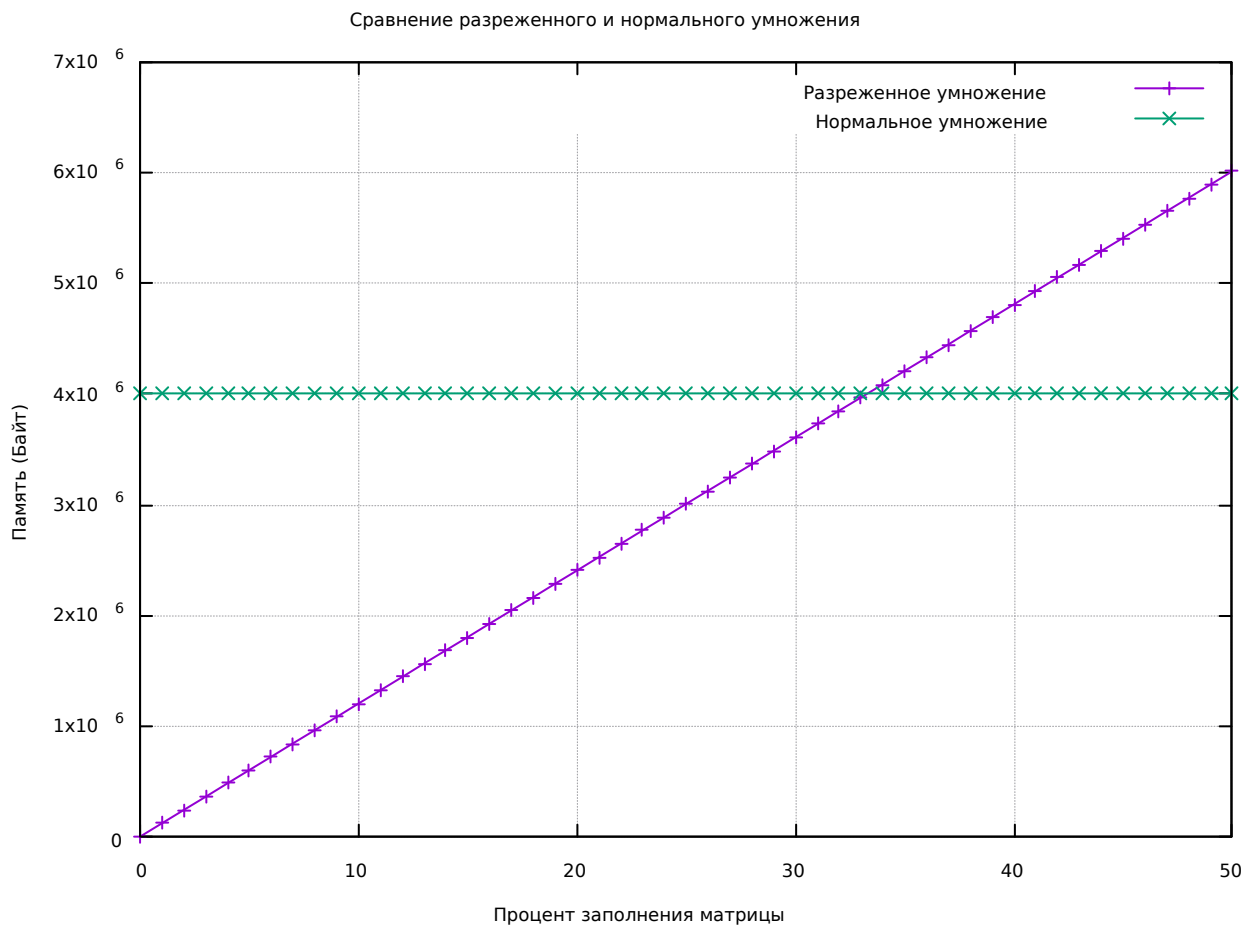


Рисунок 3: 1000x1000 память

Как видно, эффективность по времени умножения матриц в разреженном виде выше при заполнении матриц до 7 - 8%, а по памяти — до 33%.

Контрольные вопросы

1. Что такое разреженная матрица, какие способы хранения вы знаете?

Разреженная матрица — матрица, содержащая большое кол-во нулевых элементов. Хранить можно в виде обычной матрицы, с помощью линейных связанных списков, кольцевых связанных списков, двунаправленных стеков и очередей.

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

Под обычную матрицу выделяется $N \times M$ ячеек памяти. Память под разреженную матрицу выделяется в зависимости от схемы хранения, объем занимаемой памяти зависит от количества ненулевых элементов.

3. Каков принцип обработки разреженной матрицы?

Разреженные матрицы содержат большое кол-во нулей, хранятся они в структурах, хранящих только ненулевые элементы. Поэтому алгоритмы обработки оперируют лишь значащими данными, что даёт выигрыш по памяти и скорости по сравнению с алгоритмами обработки обычных матриц.

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

Разреженность матрицы эффективнее, если игнорирование нулевых элементов даёт выигрыш в производительности. При достижении определенного процента заполнения ненулевыми элементами происходит значительное падение эффективности по времени.

Вывод

Использование разреженных матриц имеет смысл только при большом количестве элементов и малом проценте содержания ненулевых элементов — иначе стандартные алгоритмы хранения и обработки будут эффективнее, как по памяти, так и по времени. Чем больше размеры матриц, тем меньше процент заполненности, необходимый для того, чтобы стандартный алгоритм стал эффективнее.