|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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)** |

ФАКУЛЬТЕТ Информатика и системы управления (ИУ)

КАФЕДРА Программное обеспечение ЭВМ и информационные технологии (ИУ7)

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

Студент Фам Минь Хиеу

Группа ИУ7 – 32Б

Преподаватель Никульшина Т.А

*2022 г.*

1. **Условие задачи**

Разработать программу умножения разреженных матриц. Предусмотреть возможность ввода данных, как с клавиатуры, так и использования заранее подготовленных данных. Матрицы хранятся и выводятся в форме трех объектов. Для небольших матриц можно дополнительно вывести матрицу в виде матрицы. Сравнить эффективность (по памяти и по времени выполнения) стандартных алгоритмов обработки матриц с алгоритмами обработки разреженных матриц при различной степени разреженности матриц и различной размерности матриц.

1. **Техническое задание**

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.

1. **Входные данные:**
2. **Целое число –** номер пункта.
3. **Командно-зависимые данные:**
4. Количество строк\столбцов матрицы
5. Элементы матрицы в формате “индекс строки индекс столбца значение элемента”
6. Количество ненулевых элементов матрицы в процентах
7. **Выходные данные:**
8. Исходные и результирующие матрицы в стандартном формате и разреженном формате
9. Сравнения времени сложения матриц разного формата
10. Сравнения занимаемой памяти матриц разного формата
11. **Функция программы:** 
    1. Ввести две матрицы с клавиатуры
    2. Автоматически генерировать две матрицы с заданным процентом разреженности
    3. Сложить две матрицы в стандартном формате
12. Сложить две матрицы в разреженном формате
13. Вывести две матрицы в обычном формате
14. Вывести две матрицы в разреженном формате
15. Сравнить время выполнения операций и объем памяти при сложении матриц в двух форматах
16. Выйти
17. **Обращение к программе:**

Запускается через терминал с помощью команды ./app.exe

1. **Аварийные ситуации:**
2. **Некорректный ввод команды.**

**На входе:** число большее 7, либо меньшее 0

**На выходе:** "Номер пункта должно быть в диопазоне [0 – 7]"

1. **Пустая матрица**

**На входе:** Вызов любой из команд 3-7 при пустых матрицах.

**На выходе:** "Матрицы пока не введены."

1. **Неверный ввод числа**

**На входе:** неверно введенный индекс строки матрицы

**На выходе:** " Матрица введена не верна. "

1. **Неверный ввод числа**

**На входе:** неверно введенный индекс столбца матрицы

**На выходе:** " Матрица введена не верна. "

1. **Неверный ввод числа**

**На входе:** неверное значение количества строк\столбцов матрицы\ненулевых элементов матрицы

**На выходе:** " Матрица введена не верна. "

1. **Неверный ввод числа**  
   **На входе:** неверное процентное значение ненулевых элементов матрицы

**На выходе:** "Размеры двух матриц должны быть равным."

1. **Описание структур данных**

За хранение матрицы в стандартном формате используются следующие переменные:

int \*\*matrix;

int n, m;

int number\_nonzero\_el;

Где:

int \*\*matrix – матрица представленная длинным массивом

int n - количество строк матрицы

int m - количество столбцов матрицы

int number\_nonzero\_el - количество ненулевых элементов матрицы

За хранение матрицы в разреженном формате используются следующие переменные:

int \*a, \*ai, \*ja;

int \*pa, \*pai, \*pja;

int number\_nonzero\_el;

Где:

int \*a, \*ai, \*ja – указатели на начало массивов

int \*pa, \*pai, \*pja – указатели на конце массивов a, ai, ja

int number\_nonzero\_el - количество ненулевых элементов матрицы

1. **Описание алгоритма**
2. Пользователю выводится меню программы
3. Пока номер пункта не равен нулю, то программа продолжается работать.
4. **Тесты**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Тест | Ввод | Результат |
| 1 | Некорректный ввод команды | -1 | Ошибка: Номер пункта должно быть в диопазоне [0 – 7] |
| 2 | Некорректный ввод команды | a | Ошибка: Номер пункта должно быть в диопазоне [0 – 7] |
| 3 | Ввод корректной команды (3-7) при пустых матрицах | любая команда 3-7 | Ошибка: Матрицы пока не введены |
| 4 | Неверное значение количества строк матрицы | -1 | Ошибка: Матрица введена не верна |
| 5 | Неверное значение количества строк матрицы | a | Ошибка: Матрица введена не верна |
| 6 | Неверное значение количества столбцов матрицы | 0.1 | Ошибка: Матрица введена не верна |
| 7 | Неверное значение количества столбцов матрицы | a | Ошибка: Матрица введена не верна |
| 8 | Неверное процентное значение ненулевых элементов матрицы | 100 | Ошибка: Размеры двух матриц должны быть равным и 0 < процентное значение < 100. |
| 9 | Неверное процентное значение ненулевых элементов матрицы | -1 | Ошибка: Размеры двух матриц должны быть равным и 0 < процентное значение < 100. |
| 10 | Неверное значение количества ненулевых элементов матрицы | 10 (при размере матрицы 3 х 3) | Ошибка: Матрица введена не верна |
| 11 | Неверное значение количества ненулевых элементов матрицы | -1 | Ошибка: Матрица введена не верна |
| 12 | Неверное значение номера строки | -1 | Ошибка: Матрица введена не верна |
| 13 | Неверное значение номера строки | 6 (при размере матрицы 5 х 5) | Ошибка: Матрица введена не верна |
| 14 | Неверное значение номера столбца | -6 | Ошибка: Матрица введена не верна |
| 15 | Неверное значение номера столбца | 6 (при размере матрицы 5 х 5) | Ошибка: Матрица введена не верна |
| 16 | Неверное значение элемента матрицы | a | Ошибка: Матрица введена не верна |
| 17 | Неверное значение элемента матрицы | 1.1 | Ошибка: Матрица введена не верна |
| 18 | Ввод матриц вручную (пункт 1) | корректно введенные матрицы | Заполненные матрицы |
| 19 | Заполнение матриц случайными числами (пункт 2) | корректные размерности матриц и процент разреженности матриц | Матрицы заполненные случайными элементами |
| 20 | Сложение матриц в стандартной форме (пункт 3) |  | Результирующая матрица, полученная поэлементным сложением 1 и 2 матриц |
| 21 | Сложение матриц в разреженном столбцовом формате (пункт 4) |  | Результирующая матрица, полученная поэлементным сложением 1 и 2 матриц |
| 22 | Вывод  исходных матриц в стандартном формате на экран (пункт 5) |  | Исходные матрицы в стандартном формате |
| 23 | Вывод  исходных матриц в столбцовом формате на экран (пункт 6) |  | Исходные матрицы в разреженном формате |
| 24 | Вывести сравнение времени сложения и объема занимаемой памяти обычной и разреженной матриц (пункт 7) |  | Вывод сравнения времени сложения и объема занимаемой памяти матриц разных форматов |
| 25 | Выход из программы (пункт 0) |  | Завершение работы программы |

1. **Оценка эффективности**

Измерения эффективности сложения матриц в разных форматах будут производиться в секундах.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Процент разреженности | tобычной формы | tразреженной формы | Vобычной формы | Vразреженной формы |
| 100x100 | 1% | 0.000059 | 0.000007 | 40000 | 1980 |
| 3% | 0.000044 | 0.000018 | 40000 | 5140 |
| 5% | 0.000045 | 0.000035 | 40000 | 8268 |
| 10% | 0.000049 | 0.000092 | 40000 | 15628 |
| 15% | 0.000047 | 0.000196 | 40000 | 22540 |
| 500x500 | 1% | 0.001182 | 0.000176 | 1000000 | 41732 |
| 3% | 0.001237 | 0.000904 | 1000000 | 120308 |
| 5% | 0.001167 | 0.002238 | 1000000 | 196892 |
| 10% | 0.001244 | 0.005419 | 1000000 | 382340 |
| 15% | 0.001176 | 0.011904 | 1000000 | 555932 |
| 1000x1000 | 1% | 0.004741 | 0.000617 | 4000000 | 163292 |
| 3% | 0.004038 | 0.003715 | 4000000 | 476612 |
| 5% | 0.005559 | 0.011161 | 4000000 | 784004 |
| 10% | 0.003196 | 0.037613 | 4000000 | 1524164 |
| 15% | 0.004341 | 0.077995 | 4000000 | 3205876 |

1. **Контрольные вопросы**

**1. Что такое разреженная матрица, какие схемы хранения таких матриц вы знаете?**

Разреженная матрица — это матрица, содержащая большое количество нулей. Схемы хранения матрицы: связанная схема хранения, строчный формат, столбовой формат, кольцевой связанный список, двунаправленные стеки и очереди.

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

Под обычную матрицу (N – количество строк, M – количество столбцов) выделяет N \* M \* sizeof(тип элементов).

Для разреженной матрицы количество ячеек памяти зависит от способа. В случае разреженного формата требуется (К + M) \* sizeof(int) + K \* sizeof(тип элементов) байтов, где К — количество ненулевых элементов, M – количество столбцов.

**3. Каков принцип обработки разреженной матрицы?**

Алгоритмы обработки разреженных матриц предусматривают работу только с ненулевыми элементами.

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

Стандартные алгоритмы обработки матриц выгоднееприменять при большом количестве ненулевых элементов (начиная примерно с 4%). Зависит от количества ненулевых элементов и размера матриц.

Стоит отметить, что если расход памяти в программе не так важен, но важно время выполнения программы, то при большом количестве ненулевых элементов лучше воспользоваться стандартным алгоритмом, и разряженного в случае небольшого количества ненулевых элементов.

1. **Вывод**

Разреженные матрицы удобнее использовать при небольшом количестве нулей (при ~3-4% ненулевых элементов) в матрице, так как тратится меньший объем памяти (примерно в 2-4 раза меньше стандартного представления), а так же время выполнения программы значительно меньше (примерно в 1,5-2 раза), чем при использовании стандартного алгоритма умножения.

Но при >4% ненулевых элементов и при большом размере матриц выгоднее использовать стандартный алгоритм, так как время становится таким же или меньше, и если важна память, то так же при 4% ненулевых элементом выгоднее использовать стандартный алгоритм, так как затраты по памяти значительно меньше чем в виде разреженном.

Стоит отметить, что когда процент ненулевых элементов матрицы не превышает 3-4%, использование алгоритмов работы с разреженными матрицами будет выигрывать как по памяти, так и по времени.