|  |  |
| --- | --- |
| **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»**

Название предмета: Типы и структуры данных

Студент: Малышев Иван Алексеевич

Группа: ИУ7-31Б

*2020 г.*

**Описание условия задачи**

Разреженная (содержащая много нулей) матрица хранится в форме 3-х объектов:

* вектор **A** содержит значения ненулевых элементов;
* вектор **JA** содержит номера столбцов для элементов вектора **A**;
* связный список **IA**, в элементе **Nk** которого находится номер компонент в **A** и **JA**, с которых начинается описание строки **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 **4**.
2. **Командно-зависимые данные:**

* количество строк/столбцов матрицы, количество элементов матрицы, элементы матрицы в формате **“индекс строки индекс столбца – значение элемента”**;
* количество строк/столбцов матрицы, процент ненулевых элементов матрицы.
* имя файла, содержащего информацию о матрице в разряженном виде.

**Выходные данные:**

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

**Ограничения:**

* Количество информации о разреженной матрице всегда корректно. Ответственность за неправильную обработку разреженных матриц из файла возлагается на пользователя.
* Вывод матриц происходит сразу после сложения матриц.
* Вывод матриц в обычном виде возможен до 30 столбцов и до 30 строк, в ином случае выводится информация о разреженности.
* Элементы матрицы – действительные числа.
* Максимальный размер матрицы: ограничено ОС.

**Функция программы:** программа выполняет ряд функций, предлагаемых пользователю в процессе её работы. Она позволяет:

1. Ввести с клавиатуры и сложить две разреженные матрицы
2. Сгенерировать и сложить две разреженные матрицы
3. Считать две разреженные матрицы из файлов и сложить их
4. Сравнить время выполнения операций и объем памяти

**Обращение к программе:** запускается из терминала ОС по имени «app.exe».

**Аварийные ситуации:**

1. Некорректный ввод номера команды.
2. Некорректный ввод количества строк или столбцов матрицы.
3. Некорректный ввод имени файла, содержащего информацию о разреженной матрице.
4. Некорректный ввод процента заполнения разреженной матрицы.
5. Некорректный ввод информацию о разреженной матрице из файла.

**Структуры данных**

За стандартное хранение матрицы отвечает именованная структура **default\_matrix\_t**, описанная как:

**typedef struct**

**{**

***double* \*\*matrix;**

***int* n;**

***int* m;**

**} default\_matrix*\_t*;**

Поля структуры:

* ***int* \*\*matrix –** указатель на массив элементов типа int;
* ***int* n *–***количество строк матрицы;
* ***int* m *–*** количество столбцов матрицы.

За хранение матрицы в разреженном виде отвечает именованная структура **matrix\_t**, описанная как:

**typedef struct**

**{**

***int n;***

***int m;***

***int nz\_elems;***

***int \*a;***

***int \*ja;***

***int \*ia;***

**} matrix\_t;**

Поля структуры:

* ***int* n *–***количество строк матрицы;
* ***int* m *–*** количество столбцов матрицы;
* ***int nz\_elems –***количество ненулевых элементов в матрице;
* ***int \*ja*** *–* массив, который содержит номера столбцов для ненулевых элементов;
* ***int \*ia –*** массив, который содержит номера ненулевых элементов, с которых начинается описание очередной строки;
* ***int \*a –*** массив ненулевых элементов.

**Описание основных подпрограмм**

**int input\_matrix(matrix\_t \*matrix)** – ввод разреженной матрицы

**void output\_matrix(const matrix\_t \*matrix)** – вывод разреженной матрицы

**void generate\_matrix(matrix\_t \*matrix, double persent)** – генерация разреженной матрицы

**int fread\_matrix(matrix\_t \*matrix)** – чтение разреженной матрицы из файла

**int add\_matrixes(matrix\_t \*m1, matrix\_t \*m2, matrix\_t \*res)** – сложение разреженных матриц

**void free\_matrix(matrix\_t \*matrix)** – очистка памяти от разреженных матриц

**int create\_default\_matrix(default\_matrix\_t \*dm)** – создание стандартной матрицы

**default\_matrix\_t \*add\_matrixes\_default(default\_matrix\_t \*m1, default\_matrix\_t \*m2)** – сложение стандартных матриц

**void free\_default\_matrix(default\_matrix\_t \*matrix)** – очистка памяти от стандартных матриц

**void generate\_default\_matrix(default\_matrix\_t \*matrix, double persent)** – генерация стандартной матрицы

**void measuring\_algorithms\_time(void)** – сравнение времени выполнения и объема памяти

**Тесты**

Положительные тесты

|  |  |  |  |
| --- | --- | --- | --- |
|  | Тест | Пользовательский ввод | Результат |
| 1 | Обычный тест  (квадратные матрицы) | 3 3 3 3  4 4  1 0 2 1 0 3  0 3 0 5 4 0  4 0 0 0 0 2 | 3 3  6  2 0 5  5 8 0  4 0 2 |
| 2 | Имеются пустые строки | 2 2 2 2  1 1  0 0 1 0  0 1 0 0 | 2 2  2  1 0  0 1 |
| 3 | Нет сложения | 3 3 3 3  5 4  1 0 3 0 2 0  0 5 0 4 0 6  7 0 9 0 8 0 | 3 3  9  1 2 3  4 5 6  7 8 9 |
| 4 | Обычный тест | 2 3 2 3  4 3  1 0 2 4 3 0  4 0 3 0 0 1 | 2 3  5  5 3 2  4 0 4 |

Негативные тесты

|  |  |  |  |
| --- | --- | --- | --- |
|  | Тест | Пользовательский ввод | Результат |
| 1 | Некорректный ввод команды | 7 | Неизвестная операция |
| 2 | Некорректный ввод количества строк | 0 2 | Ошибка ввода размера матрицы. Попробуйте снова: |
| 3 | Некорректный ввод количества строк | A 2 | Ошибка ввода размера матрицы. Попробуйте снова: |
| 4 | Некорректный ввод количества столбцов | 2 -2 | Ошибка ввода размера матрицы. Попробуйте снова: |
| 5 | Некорректный ввод количества столбцов | 2 B | Ошибка ввода размера матрицы. Попробуйте снова: |
| 6 | Некорректный ввод  количества ненулевых элементов | 3 3  0 | Ошибка ввода количества ненулевых элементов. Попробуйте снова: |
| 7 | Некорректный ввод индексов элемента | 3 3  4  -1 1 | Введены некорректные индексы. Попробуйте снова: |
| 8 | Введён нулевой элемент | 2 2  4  0 0  0 | Введён некорректный элемент. Попробуйте снова: |
| 9 | Введены матрицы разных размеров | 2 2 3 2  … … | Размеры матриц не совпадают |
| 10 | Некорректный ввод имени файла | A | Ошибка загрузки |
| 11 | Некорректный ввод имени файла |  | ОШИБКА: Пустой ввод. Повторите попытку. |
| 12 | Некорректный ввод из файла | Имя файла  Данные из файла | Сообщение о типе ошибки при чтении информации из файла. |
| 13 | Некорректный ввод процента заполненности матрицы | 1.01 | Ошибка загрузки |

**Оценка эффективности**

Измерения эффективности способов сложения будут производиться в единицах измерения – тактах процессора. При записи результатов использовалось среднее количество тактов, полученное по результатам 100 измерений.

**Время сложения:**

**5% заполнения**

|  |  |  |
| --- | --- | --- |
| Размеры | Разреженная матрица | Обычная матрица |
| 100х100 | 49287 | 320230 |
| 250х250 | 269217 | 1911049 |
| 500х500 | 1077848 | 7640022 |

**30% заполнения**

|  |  |  |
| --- | --- | --- |
| Размеры | Разреженная матрица | Обычная матрица |
| 100х100 | 240952 | 325643 |
| 250х250 | 1408900 | 1836603 |
| 500х500 | 5781614 | 7924126 |

**45% заполнения**

|  |  |  |
| --- | --- | --- |
| Размеры | Разреженная матрица | Обычная матрица |
| 100х100 | 417627 | 373680 |
| 250х250 | 2459968 | 2242149 |
| 500х500 | 8296250 | 8397764 |

**50% заполнения**

|  |  |  |
| --- | --- | --- |
| Размеры | Разреженная матрица | Обычная матрица |
| 100х100 | 514638 | 432793 |
| 250х250 | 2912569 | 2446874 |
| 500х500 | 9256793 | 8504895 |

**Объём занимаемой памяти (в байтах):**

В качестве объема памяти берётся объём занимаемый результирующей матрицей.

**5% заполнения**

|  |  |  |
| --- | --- | --- |
| Размеры | Разреженная матрица | Обычная матрица |
| 100х100 | 12144 | 80008 |
| 250х250 | 74112 | 500050 |
| 500х500 | 295216 | 2000200 |

**30% заполнения**

|  |  |  |
| --- | --- | --- |
| Размеры | Разреженная матрица | Обычная матрица |
| 100х100 | 61560 | 80008 |
| 250х250 | 383976 | 500050 |
| 500х500 | 1532308 | 2000200 |

**40% заполнения**

|  |  |  |
| --- | --- | --- |
| Размеры | Разреженная матрица | Обычная матрица |
| 100х100 | 76884 | 80008 |
| 250х250 | 480240 | 500050 |
| 500х500 | 1923616 | 2000200 |

**45% заполнения**

|  |  |  |
| --- | --- | --- |
| Размеры | Разреженная матрица | Обычная матрица |
| 100х100 | 83880 | 80008 |
| 250х250 | 525144 | 500050 |
| 500х500 | 2095588 | 2000200 |

**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 N \* M ячеек памяти, где N – строки, а M – столбцы. Для разреженной матрицы – зависит от способа. В случае разреженного строчного формата, требуется 2 \* K + N + 1 ячеек памяти, где K – количество ненулевых элементов.

**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 50%). Стоит отметить, что если не так важна память, занимаемая матрицами, но важно время, то в случае сложения лучше воспользоваться алгоритмами сложения разреженных матриц.

**Вывод**

Использование алгоритмов хранения и обработки разреженных матриц в случае сложения выгодно при малом количестве ненулевых элементов, примерно до 50% заполненности матриц. Хранение матриц в разреженном формате проигрывает по памяти в случаях, когда она заполнена на 45% и далее, так как структура для хранения разреженных матриц довольно нагружена полями для хранения различных компонент матриц. Помимо занимаемой памяти, разреженные матрицы гораздо дольше обрабатываются при заполненности матриц от 50%, так как требуется много времени для составления портрета матрицы.