Отчет по 3 лабораторной работе

По дисциплине «Типы и структуры данных»

Подготовил Жабин Дмитрий

Группа ИУ7-34Б

Вариант 2

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

**Задача**

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.

**Внешняя спецификация программы.**

**Входные данные:** две матрицы (размеры матрицы - пара натуральных чисел, не превосходящих 1000; в случае выбора ввода с клавиатуры - набор значений - целых чисел, не превосходящих 1000 по модулю, в случае выбора ввода из файла - имя файла (с расширением), содержащего набор значений с теми же ограничениями, в случае выбора автоматического заполнения матрицы - процент заполнения нулями (целое число от 0 до 100)).

**Выходные данные:** матрица такого же формата - результат сложения двух данных матриц, разреженное представление (в форме 3-х объектов) каждой из трех матриц. Информация об эффективности обработки  
матриц со стандартным и разреженным способами хранения.

**Задача программы:**

Произвести операцию сложения двух матриц, хранящихся в стандартной форме - в одном случае, и в разреженной - в другом. Сравнить затраты по времени и памяти в этих случаях.

**Способ обращения к программе:** Запуск программы в консоли (./app.exe).

**Возможные ошибки**

1) Некорректный ввод размеров или значений матрицы

Сообщение “Input error”, программа завершает работу.

2) Ошибки выделения памяти

Сообщение "Memory error”, программа завершает работу.

**Структуры данных.**

Для хранения матриц в разреженном виде выбрана следующая структура:

typedef struct

{

int \*MN; - массив ненулевых элементов матрицы

int \*IM; - массив номеров строк для элементов массива MN

int \*JM; - массив индексов в MN первых ненулевых элементов столбцов матрицы

} sparse\_M;

Для хранения матриц в стандартном виде выбран двумерный массив:

int \*\*A = NULL, \*\*B = NULL, \*\*S = NULL;

**Функции**

int input\_matrix(int \*\*\*matrix, int n, int m); - функция ввода матрицы с клавиатуры

int input\_matrix\_from\_file(int \*\*\*matrix, int n, int m); - функция ввода матрицы из файла

int generate\_matrix(int \*\*\*matrix, int n, int m); - функции автоматического заполнения матрицы

void print\_matrix(int \*\*matrix, int n, int m); - функция вывода матрицы на экран

void print\_sparse\_matrix(sparse\_M sparse\_A, int len, int m); - функция вывода разреженной матрицы на экран

void add\_up\_matrix(int \*\*A, int \*\*B, int \*\*S, int n, int m); - функция сложения матриц в стандартном виде

void add\_sparse\_matrix(sparse\_M matrix\_a, sparse\_M matrix\_b, sparse\_M \*res, int m, int lenA, int lenB, int \*lenS); - функция сложения матриц в разреженном виде

int create\_sparse\_matrix(int \*\*A, int n, int m, sparse\_M \*sparse\_A, int \*lenA); - функция создания матрицы в разреженном виде

**Алгоритм**.

При сложении матриц в стандартном виде происходит попарное сложение элементов исходных матриц, находящихся в одинаковых позициях (в одних и тех же строке и столбце), их сумма помещается в ту же позицию в матрицу-результат.

При сложении матриц в разреженном виде:  
Создаются счетчики a\_ind, b\_ind, res\_ind ненулевых элементов в первой, второй матрице и матрице-результате (инициализированы нулями). Организован цикл перебора столбцов матрицы (i изменяется от 0 до m-1).

В цикле:

i-ому элементу массива JA матрицы-результата присваивается текущее значение res\_ind (количество ненулевых элементов в уже просмотренных столбцах матрицы).

Перебираются ненулевые элементы в текущем (i-ом) столбце матриц-слагаемых:

1) Если они еще не закончились в обеих матрицах, то сравниваются номера строк, в которых они расположены: если номера совпали, то в массив A матрицы-результата записывается сумма этих элементов, в массив IA записывается номер их строк, счетчики a\_ind и b\_ind увеличиваются на 1; если же номера строк отличаются, в матрицу-результат записывается элемент, находящийся в столбце выше (номер строки которого меньше) - в массив A матрицы-результата записывается его значение, в массив IA - номер строки, соответствующий счетчик увеличивается на 1.

2) Если же при переборе в одной из матриц в столбце закончились ненулевые элементы, то в матрицу-результат записывается элемент из второй матрицы, соответствующий счетчик увеличивается на 1.

Во всех перечисленных случаях при добавлении элемента в матрицу-результат счетчик res\_ind увеличивается на 1.

**Тесты.**

**Негативные**

|  |  |
| --- | --- |
| Ввод | Вывод |
| Input numbers of matrix rows and columns (both from 1 to 1000): 5 1001 | Input error |
| Input numbers of matrix rows and columns (both from 1 to 1000): 3 3  Input matrix elements:  1 0 0  0 -2 3  ab 5 0 | Input error |
| Input numbers of matrix rows and columns (both from 1 to 1000): 3 3  Input matrix elements:  1 0 0  0 -2 3.75  0 5 0 | Input error |

**Позитивные**

|  |  |  |  |
| --- | --- | --- | --- |
| Матрица 1 | Матрица 2 | Матрица суммы в стандартном виде | Матрица суммы в разреженном виде |
| 1 0 0 0  0 2 0 0  0 -3 0 0 | 0 6 0 0  0 -2 7 0  10 -2 0 1 | 1 6 0 0  0 0 7 0  10 -5 0 1 | N: 1 10 6 -5 7 1  J: 0 2 0 2 1 2  I: 0 2 4 5 |
| 0 5 0 0  3 0 0 0  -1 0 0 2  2 1 0 -6 | 1 3 0 0  0 7 0 -1  2 0 0 4  -2 3 0 6 | 1 8 0 0  3 7 0 -1  1 0 0 6  0 4 0 0 | N: 1 3 1 8 7 4 -1 6  J: 0 1 2 0 1 3 1 2  I: 0 3 6 6 |
| 0 0  1 0 | 0 0  -1 0 | 0 0  0 0 | N:  J:  I: 0 0 |

**Оценка эффективности**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Размерность матриц | Процент заполнения нулями | Время, тактов процессора | | Память, байт | |
| Стандартный | Разреженный | Стандартный | Разреженный |
| 200\*200 | 40 | 514904 | 1838524 | 484800 | 655368 |
| 60 | 1175320 | 4436550 | 463024 |
| 80 | 1223880 | 2109962 | 245504 |
| 87 | 1361540 | 1185428 | 163328 |
| 95 | 1133130 | 498580 | 65552 |
| 500\*500 | 40 | 2243202 | 10704466 | 3012000 | 4084472 |
| 60 | 2119278 | 17175106 | 2885024 |
| 80 | 4241530 | 4935768 | 1526728 |
| 88 | 3319544 | 2763616 | 936952 |
| 95 | 3963056 | 1710018 | 400952 |
| 1000\*1000 | 40 | 8995044 | 47904844 | 12024000 | 16331264 |
| 60 | 18158608 | 38173170 | 11533672 |
| 80 | 16312546 | 15487346 | 4631040 |
| 95 | 9180272 | 4087498 | 1591912 |

**Вывод**

Алгоритм сложения матриц в разреженном виде работает быстрее при сильном разрежении (не менее 80-85%) элементов матриц. Его использование позволяет экономить память при заполнении нулями более, чем на 50%. В противном случае быстрее работает стандартный алгоритм сложения матриц. Если матрица не разреженная, то алгоритм сложения матриц в разреженном виде работает очень долго, потому что каждую строку надо долго обрабатывать, в отличие от стандартного алгоритма, в котором нужно просто пройти по каждому элементу.

**Контрольные вопросы**

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

Разреженная матрица – матрица, содержащая достаточно большое количество элементов, из которых лишь малая часть является ненулевыми (n^(1+g) для матрицы размерности n, g<1).

Простейшая схема хранения разреженной матрицы: хранить массив ненулевых элементов (AN), и два массива их «координат» (I, J).

Кнут предложил хранить дополнительно массивы NR (содержит номер из AN следующего ненулевого j элемента, расположенного в матрице по строке) и NC (по столбцу), а также массивы JR и JC (указатели для входа в строку и столбец).

Чанг и Густавсон предложили схему разреженного строчного формата: значения ненулевых элементов хранятся в массиве AN, соответствующие им столбцовые индексы - в массиве JA. Кроме того, используется массив указателей IA, отмечающих позиции AN и JA, с которых начинаются описание очередной строки. Дополнительная компонента в IA содержит указатель первой свободной позиции в JA и AN.

Аналогично устроен разреженный столбцовый формат.

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

Для хранения обычной матрицы: M \* N \* sizeof(elem\_t).

Память под разреженную матрицу выделяется в зависимости от схемы хранения. Память выделяется по мере наполнения ненулевыми элементами.

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

Принцип предполагает нахождение элементов по индексам, или по отношению к другим элементам. Работа осуществляется только с ненулевыми элементами.

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

Чем больше ненулевых элементов в матрице, тем менее эффективно использовать разреженные алгоритмы. При достижении определенного процента наполнения наблюдается значительная деградация разреженного алгоритма по времени работы.