**МГТУ им. Н.Э.Баумана**

**Дисциплина “Типы и структуры данных”**

**Отчет по Лабораторной работе №3 по теме “Обработка разреженных матриц”.**

Работу выполнил:

Студент: Зейналов Зейнал

ИУ7-31Б

Варинат 14(3)

Москва, 2018.

*Описание задачи:*

*Разработать программу сложения разреженных матриц. Предусмотреть возможность ввода данных, как с клавиатуры, так и использования автоматического заполнения. Матрицы хранятся и выводятся в форме трех объектов. Для небольших матриц можно дополнительно вывести матрицу в виде матрицы. Величина матриц - любая. 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.*

*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 умножения матрицы и вектора-столбца, хранящихся в этой форме, с получением результата в той же форме.*

*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.*

*Описание ТЗ*

*Введение*:

*Наименование программы* - “Мультипликатор разреженных матриц”.

Программа предназначена к применению в учебных целях.

*Основания для разработки*:

Основанием для проведения разработки является Лабораторная работа № 3, выданная в МГТУ им. Н.Э. Баумана в учебных целях.

*Назначение разработки*:

*Функциональное назначение программы*: предоставление пользователю возможностей умножения матриц в обычном виде, в разреженном виде и сравнения эффективности этих способов. *Эксплуатационное назначение программы*: программа должна использоваться в ходе проведения Лабораторной работы № 3 в МГТУ им. Н.Э. Баумана, пользователями программы должны являться преподаватели и студенты университета.

*Требования к программе или программному изделию*:

* *Требования к функциональным характеристикам*: программа должна предоставить пользователю возможность самостоятельно ввести матрицы с помощью координатного метода или заполнить матрицы автоматически, при этом пользователь должен ввести размерность матриц и степень их заполненности. Программа должна просуммировать матрицы обычным способом и способом сложения разреженных матриц, сравнить их эффективность по памяти и по времени и вывести матрицу в обычном виде, если она меньше, чем 50 на 50, и в разреженном виде. При вводе некорректных значений пользователем, программа должна выдать сообщение об этом и попросить ввести еще раз.
* *Требования к надежности*: программа должна воспринимать любой ввод и не завершаться аварийно. При вводе целочисленных значений должны проверяться введенные символы и диапазон возможных значений. При некорректном вводе пользователю должно быть выведено сообщение об этом и предложено ввести значение снова.
* *Требования к эксплуатации*: Программа не требует особых климатических условий. Программа не требует проведения каких-либо видов обслуживания. Для нормального функционирования программы достаточно одного человека - пользователя.

*Описание СД*

*Для реализации данной программы были использованы следующие структуры:*





Представление матрицы в строковом формате.



*Алгоритм:*

*Алгоритм умножения матрицы, представленной в строчном формате на матрицу – столбец.*

for (int i = 0; i < n; i ++)

{

res[i][0] = 0;

for (int j = JA[i]; j < JA[i+1]; j ++)

{

res[i][0] += A[j] \* arr[0][IA[j]];

}

}

*Набор тестов:*

* Некорректные значения

|  |  |
| --- | --- |
| Ввод | Вывод программы |
| Число строк меньше 1 | Некорректный ввод |
| Число столбцов меньше 1 | Некорректный ввод |
| Число ненулевых элементов меньше 0 или больше размера матрицы | Вы ввели неверную размерность |
| Координатный ввод: индекс больше числа строк - 1, числа столбцов -1, меньше 0 | Вы допустили ошибку, попробуйте еще раз |
| Координатный ввод: значение равное 0 | Вы ввели ноль, введите ненулевой элемент |
| Координатный ввод: повторный ввод элемента | Этот элемент уже задан, введите другой |
| Автоматический ввод: процент заполнения больше 100 или меньше 0 | Неверный ввод |

* Символьный ввод

|  |  |
| --- | --- |
| Ввод | Вывод программы |
| Любой нецелочисленный ввод | Неверный ввод, введите еще раз |

Тесты:

Вывод результирующего вектора - столбца

1. Процент заполнения нулями – 50 % (10х10)

A1 is: -2105 -582 -360 -608 241 1162 950 -1476 -86 1046

JA1 is: 0 0 0 0 0 0 0 0 0 0

IA1 is: 0 1 2 3 4 5 6 7 8 9 10

1. Процент заполнения нулями – 10% (10х10)

A1 is: -669 86 3417 -2434 -3901 -1412 3207 -4950 1623 -2740

JA1 is: 0 0 0 0 0 0 0 0 0 0

IA1 is: 0 1 2 3 4 5 6 7 8 9 10

1. Процент заполнения нулями – 100 % (10 х 10)

A1 is:

JA1 is:

IA1 is: 0

* Результаты сравнения умножения матрицы строчным методом и обычным

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Размеры матрицы | Процент заполнения матриц | Время сортировки  (в тиках процессора) | | Занимаемая память | |
| Обычная | Разреженная | Обычная | Разреженная |
| 10 | 0 | 924 | 192 | 400 | 24 |
| 25 | 919 | 662 | 200 |
| 50 | 939 | 1004 | 392 |
| 75 | 949 | 1355 | 600 |
| 100 | 920 | 1203 | 784 |
| 100 | 0 | 84468 | 6785 | 40000 | 2472 |
| 10 | 111073 | 13193 | 7176 |
| 15 | 80796 | 19405 | 11120 |
| 50 | 72327 | 54528 | 39104 |
| 100 | 87132 | 114952 | 79248 |
| 250 | 0 | 485485 | 29878 | 250000 | 14912 |
| 1 | 477353 | 37352 | 19984 |
| 5 | 427787 | 68875 | 45056 |
| 7 | 463041 | 82294 | 54840 |
| 10 | 462889 | 92219 | 65040 |
| 500 | 0 | 1838382 | 99730 | 1000000 | 59376 |
| 1 | 1815873 | 122244 | 79752 |
| 2 | 145624 | 1936250 | 98816 |
| 4 | 2080452 | 202493 | 139856 |
| 5 | 1878085 | 227819 | 159624 |

Использование алгоритма умножения разреженных матриц позволяет добиться значительного выигрыша по памяти и времени при больших размерах матрицы и малом проценте заполнения ненулевыми элементами по сравнению с обычным алгоритмом сложения.

***Вывод :*** Таким образом, можно сделать вывод, что умножения разреженных матриц представленных в строковом формате работает быстрее обычного умножения матриц при проценте разреженности до 40-45 %.

*Ответы на вопросы:*

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

Разрежённая матрица — это матрица с преимущественно нулевыми элементами.

Связная схема хранения матриц, предложенная Кнутом, предлагает хранить в массиве (например, в AN) в произвольном порядке сами элементы, индексы строк и столбцов соответствующих элементов (например, в массивах I и J), номер (из массива AN) следующего ненулевого элемента, расположенного в матрице по строке (NR) и по столбцу (NC), а также номера элементов, с которых начинается строка (указатели для входа в строку – JR) и номера элементов, с которых начинается столбец (указатели для входа в столбец - JC). Данная схема хранения избыточна, но позволяет легко осуществлять любые операции с элементами матрицы. Наиболее широко используемая схема хранения разреженных матриц – это схема, предложенная Чангом и Густавсоном, называемая: "разреженный строчный формат". В этом случае значения ненулевых элементов хранятся в массиве AN, соответствующие им столбцовые индексы - в массиве JA. Кроме того, используется массив указателей, например 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.

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

Под хранение обычной матрицы выделяется N \* M \* sizeof(тип матрицы).

Под хранение разреженной матрицы выделяется различное кол-во памяти в зависимости от способа хранения и разреженности матрицы.

*3. Каков принцип обработки разреженной матрицы?*

Принцип обработки разреженных матриц состоит в том, чтобы работать только с ненулевыми элементами и их индексами.

*4. В каком случае для матриц эффективнее применять стандартные алгоритмы обработки матриц? От чего это зависит?*Чем больше ненулевых элементов в матрице, тем эффективней применять стандартные способы обработки матриц. Выбор способа применения зависит от размерности матрицы, процента разреженности матрицы и реализации спо