МГТУ им. Баумана

Дисциплина “Типы и структуры данных”

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

Работу выполнила:

Громова В.П.

Группа ИУ7-31Б

Вариант №11

Описание задачи:

Разработать программу умножения вектора-строки на разреженную матрицу. Предусмотреть возможность ввода данных с клавиатуры, с использованием заранее подготовленных данных, хранящихся в файле и с возможностью генерации матриц заданного размера со случайным заполнением в заданном процентном соотношении ненулевых элементов (сгенерированная матрица записывается в файл). Матрицы хранятся и выводятся в форме трех объектов. Результирующий вектор-строка хранится и выводится в координатном формате. Для небольших матриц можно дополнительно ввести и вывести матрицу в виде матрицы. Величина матриц - любая (допустим, 1000\*1000). 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 .

Описание ТЗ:

Введение:

Наименование программы - “Калькулятор умножения разреженного вектора-строки на разреженную матрицу”.

Программа предназначена к применению в учебных целях.

Основания для разработки:

Выполнение задания лабораторной работы по дисциплине “Типы и структуры данных” в ходе учебного процесса в МГТУ им. Н.Э.Баумана.

Назначение разработки:

Функциональным назначением программы является предоставление пользователю возможности ввести индексы и соответствующие значения вектора-строки и матрицы в координатном и (при размерах < 5) обычном виде, загрузить данные из файла или же сгенерировать случайную матрицу с заданным размером и процентом ненулевых элементов. Также, в функциональные возможности программы входит получение результата умножения вектора строки на матрицу и получение информации о времени работы программы и затраченной и хранение данных памяти.

Эксплуатационное назначение программы: программа должна использоваться в учебных целях в МГТУ им. Н.Э.Баумана, пользователями программы должны являться сотрудники и студенты университета.

Требования к программе или программному изделию:

Требования к функциональным характеристикам:

В программе должно быть реализовано меню с возможностью выбора ввода данных. При неправильном вводе пункта меню, программа выводит соответствующее сообщение и завершается с соответствующим кодом возврата. При вводе данных с клавиатуры у пользователя сначала запрашивается размерность матрицы, далее в зависимости от полученного значения пользователю либо предлагается выбор обычного ввода или координатного ввода (при размерности меньше пяти), либо возможен только координатный ввод (в противном случае). На любом этапе ввода должен осуществляться контроль правильности вводимых значений. Все вводимые значения размерностей должны быть целочисленные. При вводе элементов допускается ввод не целых чисел.

Программа должна считывать матрицу из файла, запрашивая названия файла, при ошибке чтения программа не выполняет никаких действий. Программа должна генерировать файлы, содержащие матрицы. При этом у пользователя запрашивается размерность матрицы и процент заполнения (процент ненулевых элементов). Далее происходит чтение данных из созданного файла. Все описанное для матриц должно выполняться и при вводе вектора-строки. После успешно полученных данных программа должна произвести умножение вектора-строки на матрицу. При этом умножение должно быть произведено двумя способами: с использованием обычного алгоритма умножения матриц и с использованием алгоритма умножения разреженных матриц. Далее программа должна вывести результат работы на экран: при размерности меньше пяти, результирующий вектор выводится в двух форматах (обычном и координатном), при размерности больше пяти, формат вывода - координатный. Также на экране должно быть время выполнения умножения обычным алгоритмом и время умножения алгоритмом разреженных матриц. Время измеряется в тактах. К тому же должно быть выведено количество памяти, которое потребовалось для хранения матрицы и вектора-строки в обычном формате и в формате трех (для вектора-строки - двух) массивов.

Требования к надежности:

Программа должна воспринимать любой ввод и не завершаться аварийно. Если при ручном вводе для размерности массива или вектора-строки было введено не целое число или число меньшее 1, или если при ручном вводе для элемента массива или вектора-строки было введено не число, то программа выводит сообщение об ошибке и завершает работу с соответствующем кодом возврата. Если при вводе процента заполнение матрицы введено не число или число меньшее 1 или большее 100, то программа выводит сообщение об ошибке и завершается. При ошибке чтения файла, программа сообщает об этом пользователю и предлагает завершает работу с соответствующем ошибке кодом возврата. Если для заданных вектора-строки и матрицы умножение выполнить нельзя, то программа выдает соответствующее сообщение и завершает работу с кодом возврата, соответствующем ошибке.

Требования к эксплуатации: программа должна запускаться с лицензионной ОС, которая поддерживает исполняемые файлы формата .exe. Программа не требует проведения каких-либо видов обслуживания. Для нормального функционирования программы достаточно одного человека - пользователя.

Стадии и этапы разработки:

Стадии: получение задания на лабораторную работу, рабочий проект, сдача лабораторной работы.

Этапы: разработка программы, тестирование программы, разработка программной документации.

Информационные структуры и методы решения:

Для того, чтобы хранить элементы матрицы и вектора-строки в памяти программы, были выбраны массивы. Матрица хранится в программе в двух видах: в виде динамического двумерного массива и в виде трех динамических одномерных массивов, где первый массив состоит из ненулевых элементов матрицы, второй содержит номера строк, в которых находятся эти массивы и третий содержит индексы тех элементов из первого массива, которые являются первыми в столбцах. Вектор-строка хранится в матрице с помощью одного динамического массива или с помощью двух динамических массивов, где в первом массиве хранятся ненулевые элементы, а во втором - их индексы в общем массиве.

Внутренние структуры данных:

* тип \*\*double для хранения указателя на матрицу - 8 байт

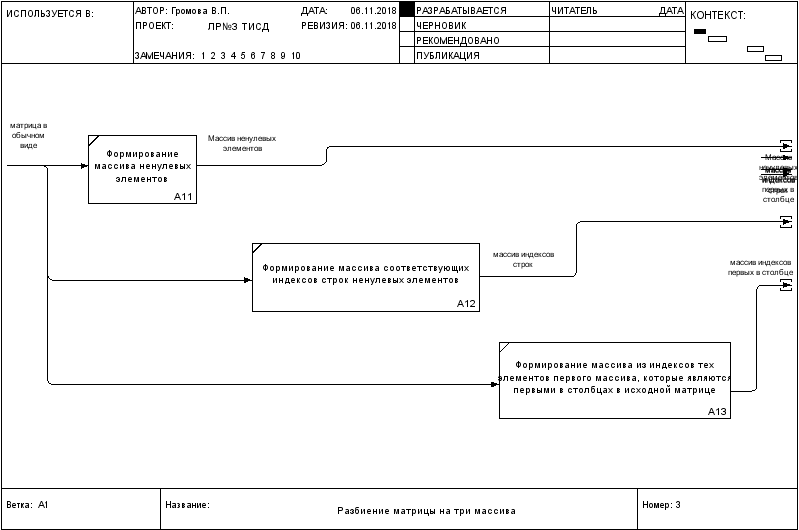
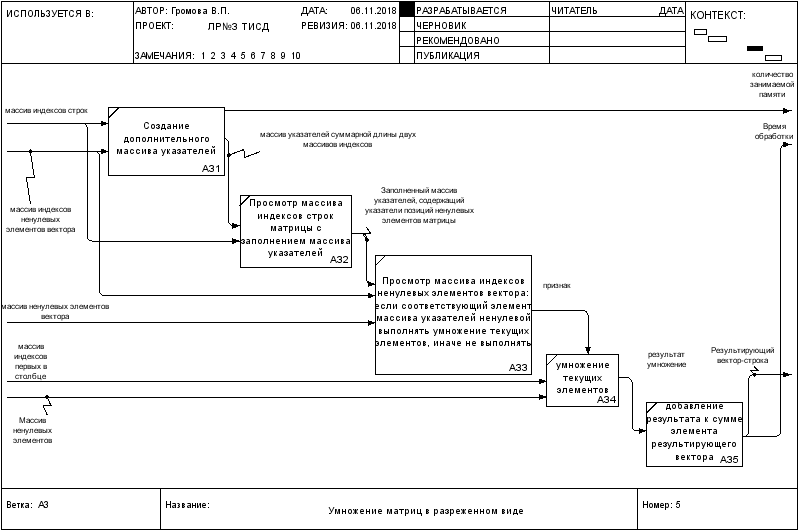
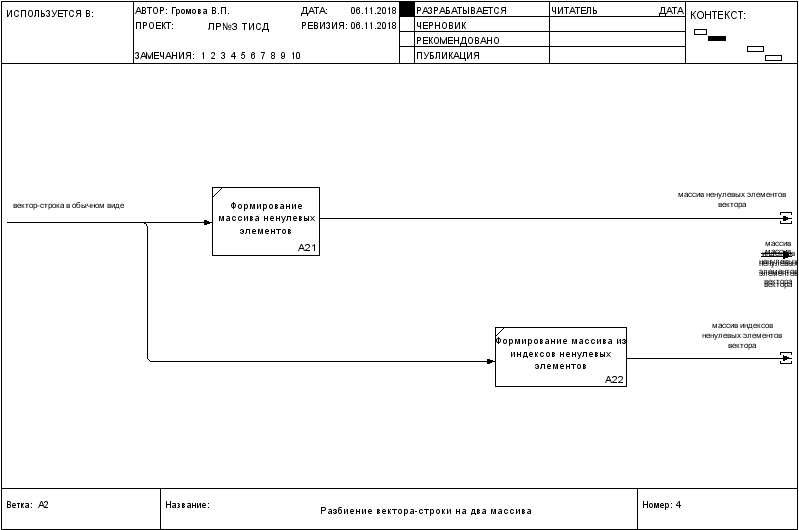
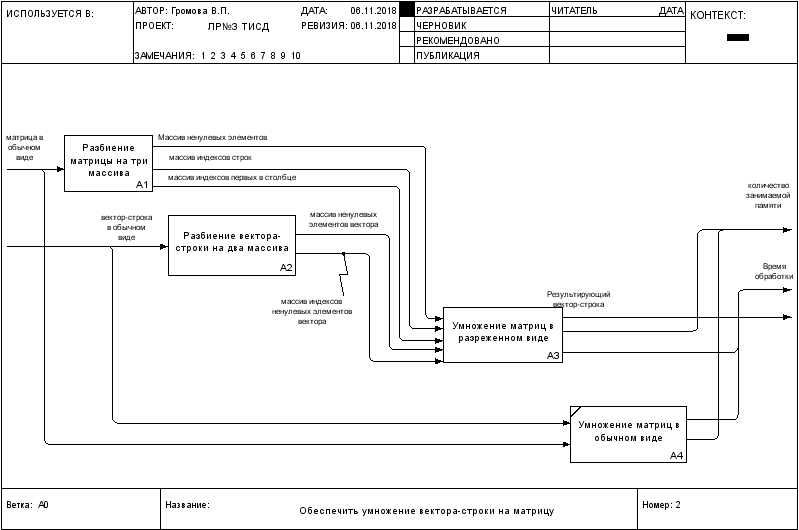
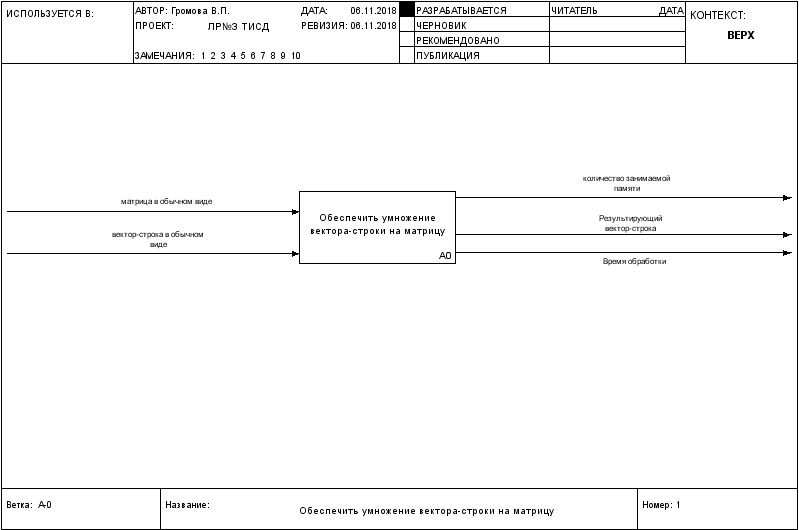
double \*\*matrix

* матрица с элементами типа double - n \* m \* 8 байт, где n и m - размерности матрицы.
* тип \*double для хранения указателя на вектор-строку - 8 байт

double \*row\_vector

* массив с элементами типа double - n \* 8 байт, где n - размерность вектора-строки
* переменные типа int для хранения размерностей матрицы и вектора-строки - 4 \* 3 байта
* массивы с элементами типа double для хранения ненулевых элементов матрицы или вектора-строки - n \* 8 байт, где n - количество ненулевых элементов.
* массивы типа int для хранения индексов строк ненулевых элементов матрицы или индексов ненулевых элементов вектора-столбца - n \* 4 байт, где n - количество ненулевых элементов.
* массив типа int для хранения индексов элементов, которые являются первыми в столбцах - 4 \* (n + 1) байт, где n - количество столбцов матрицы.
* переменная типа double для хранения считанного элемента матрицы или вектора-строки.

Описание алгоритма:



Тестовые данные:

|  |  |  |
| --- | --- | --- |
| Что проверяется | Ввод | Вывод |
| Корректная обработка выбора пункта меню | (допустимый ввод 1-3) |  |
|  | 0 | Incorrect input. |
|  | 4 | Incorrect input. |
|  | 1 | (корректно, выбор дальнейших действий) |
|  | 2 | (корректно, выбор дальнейших действий) |
|  | 3 | (корректно, выбор дальнейших действий) |
| Обработка ошибочных ситуаций при ручном вводе размерностей | (допустимый ввод: целое число больше 1) |  |
|  | line | Input error. |
|  | 2.5 | Input error. |
|  | -5 | Size should be > 1. |
|  | 0 | Size should be > 1. |
| Обработка ошибочных ситуаций при ручном вводе элементов | (допустимый ввод:число) |  |
|  | line | Incorrect input |
| Обработка ошибочных ситуаций при вводе процентного заполнения | (допустимый ввод: целое число больше 0 и меньшее или равное 100) |  |
|  | -5 | Invalid parameters. |
|  | 0 | Invalid parameters. |
|  | 101 | Invalid parameters. |
|  | 3.5 | Invalid parameters. |
| Обработка ошибочных ситуаций при чтении данных из файла | (допустимый ввод: на первой строке в файле указаны размерность матрицы и количество ненулевых элементов, все числа целые положительные и больше 0).  Количество ненулевых не должно превосходить произведение размерностей.  Элементы с одинаковыми индексами и нулевые элементы отсутствуют | Something wrong with file or data in file. Try again later :) |
|  | Файл содержит символы |  |
|  | Какое-либо из чисел в первой строке отрицательное |  |
|  | Какое-либо из чисел в первой строке равно нулю |  |
|  | Третье число в строке больше произведения первых двух |  |
|  | Файл содержит только одну строку |  |
|  | Какой-либо из индексов не целое число |  |
|  | Какой-либо из индексов отрицательный |  |
|  | Какой-либо из индексов больше соответствующей указанной размерности |  |
|  | В файле больше элементов, чем указано |  |
| Обработка корректного ввода | Введен вектор-строка размера 2 и введена матрица 1\*1 с 1 ненулевым элементом | Impossible to perform a multiplication for 1 x 2 and 1 x 1 |
|  | Введен вектор-строка размера 1 (5) и введена матрица 1\*1 с 1 ненулевым элементом (3) | Программа вывела на экран результат работы в требуемом виде (результат умножения 15) |
|  | 5 6 3  3\*3 (обычный ввод):  5 0 0  6 0 0  0 0 0 | Программа вывела на экран результат работы в требуемом виде  43 |
|  | 1 2 0 0 1  5\*5 (координатный ввод):  0 0 1  0 1 2  1 0 2  2 1 -1  4 0 -5  4 1 2 | all elements in result vector are null. |
|  |  | При выводе результата также есть и информация о времени выполнения умножения и о занимаемой памяти. |

Замеры времени (в тактах):

В таблице левый столбец размерности - время для обычного алгоритма, правый столбец - время для алгоритма разреженных матриц. Процент ненулевых элементов.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| % | 10 x 10 | 10 x 10 | 100 x 100 | 100x100 | 1000x1000 | 1000x1000 |
| 10 | 1177 | 1368 | 107600 | 19899 | 21.451.769 | 1.523.987 |
| 30 | 1187 | 1729 | 109707 | 52229 | 20.990.803 | 4.410.966 |
| 50 | 1199 | 1986 | 107394 | 80093 | 20.472.407 | 7.274.608 |
| 70 | 1200 | 2211 | 108560 | 109641 | 20.430.294 | 10.266.169 |
| 90 | 1173 | 2312 | 107768 | 134698 | 19.276.483 | 12.660.515 |
| 100 | 1223 | 2606 | 108706 | 156097 | 19.804.509 | 14.485.830 |

Вывод:

Использование алгоритма разреженных матриц эффективно по памяти до заполнения ненулевыми элементами включительно до 50%. Далее хранить три массива затратнее по памяти, чем хранение матрицы. По временной эффективности алгоритм разреженных матриц лучше использовать для матриц больших размеров (100) при проценте заполнения до 50% включительно. Однако для очень больших матриц (1000) алгоритм умножения разреженных матриц по времени оказывается эффективнее при любом проценте заполнения (однако затраты на память превышают затраты для обычного алгоритма примерно в полтора раза.

Таким образом, алгоритм разреженных матриц эффективно использовать и по памяти и по времени для матриц разреженных до 50 %.

Ответы на вопросы:

1. Что такое разреженная матрица, какие схемы хранения таких матриц Вы знаете? Разреженная матрица - это матрица, содержащая большое количество нулей, число ненулевых элементов при матрице порядка n может выражаться как где g < 1. Такие матрицы можно хранить по схеме Кнута или упрощенной схеме Чанга в разреженном строчном или в разреженном столбцовом формате.
2. Каким образом и сколько памяти выделяется под хранение разреженной и обычной матрицы? Для хранения разреженной матрицы выделяется три массива. Первые два содержат количество элементов равное количеству ненулевых элементов, а третий - количество столбцов + 1. Для хранения обычной матрицы выделяется в Си массив массивов (матрица) размерностью n x m, где n - количество строк, а m - количество столбцов.
3. Каков принцип обработки разреженной матрицы? Проходя по третьему массиву и сравнивая индексы в первых двух массивах можно дойти до каждого элемента в текущем столбце.
4. В каком случае для матриц эффективнее применять стандартные алгоритмы обработки матриц? От чего это зависит? В случаях заполнения матрицы более 50% или для матриц небольшого размера. Это зависит от количества ненулевых элементов и размерности матрицы.