|  |  |
| --- | --- |
|  | **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

КАФЕДРА Программное обеспечение ЭВМ и информационные технологии

**ОТЧЕТ ПО**

**ЛАБОРАТОРНОЙ РАБОТЕ №3**

**ПО ПРЕДМЕТУ “ТИПЫ И СТРУКТУРЫ ДАННЫХ”**

Студент Блохин Дмитрий Михайлович

Группа ИУ7-32Б

Вариант 4

Москва, 2019 г.

**Условие задачи**

Разработать программу умножения или сложения 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. 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.

**ТЗ**

Задача программы:

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

- вектор AN содержит значения ненулевых элементов;

- вектор JA содержит номера столбцов для элементов вектора AN;

- вектор IP содержит номера строк для элементов вектора AN;

1. Смоделировать операцию умножения двух матриц, 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.

**Формат ввода**

Ввод осуществляется с клавиатуры в окне консоли. Программа продолжит выполнение даже после ввода невалидных значений, пользователя вернёт в исходное меню. Для ввода новых матриц необходимо в конце заново перезапускать программу. Во время ручного заполнения матрицы или столбца необходимо сначала вводить индекс строки, затем индекс столбца и только затем значение, для прекращения ввода необходимо ввести в момент добавления нового индекса строки любое отрицательное или невалидное значение.

**Ограничения**

Нумерация строк и столбцов начинается с 0. Элементами массива могут являться только целые числа.

**Аварийный ситуации:**

При любом некорректном вводе программа возвращает пользователя в меню.

**Выходные данные:**

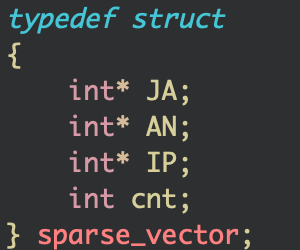
Матрица и столбец в разных вариантах хранения, а также таблица с псевдографикой для сравнения эффективности двух видов хранения матрицы.

**Описание внутренних структур данных**

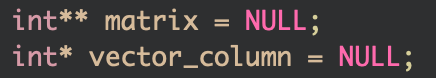
Структура для хранения разреженной матрицы:



Структура для хранения разреженного вектора:



Структура для хранения матрицы и вектора в виде массива:



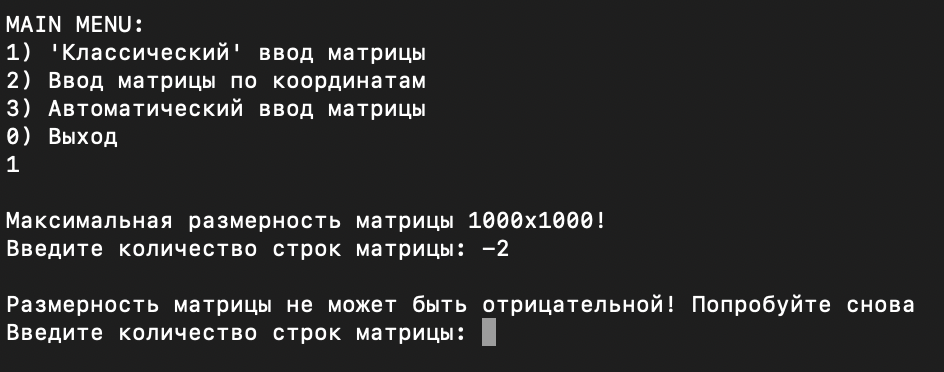
**Алгоритм**

Для умножения двух матриц я воспользовался относительно стандартным алгоритмом. Создаем пустой вектор-столбец, где элемент каждой строки является результатом умножения соответствующей строки исходной матрицы на вектор-столбец.

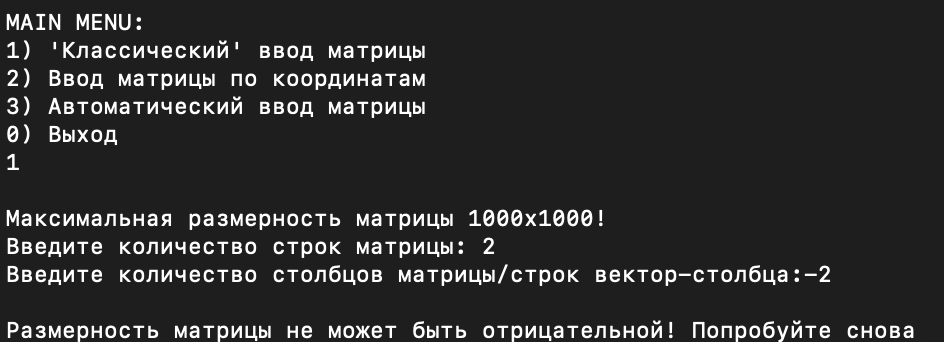
**Тесты**

**Аварийные**

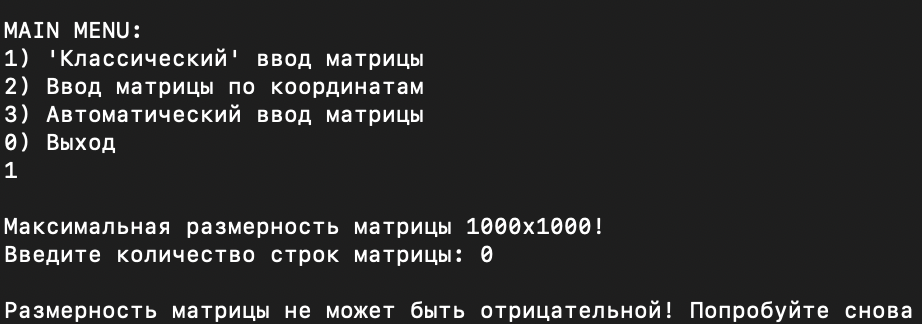
Ввод отрицательного значения в строку



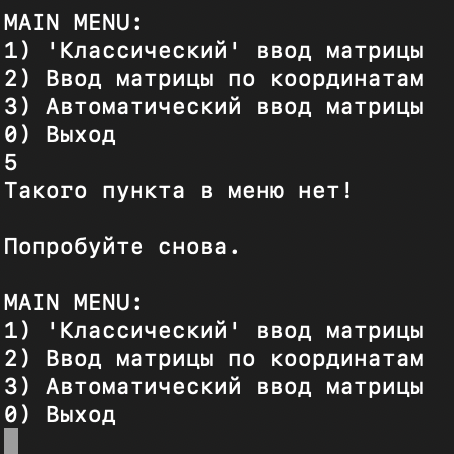
Ввод отрицательного значения в столбец



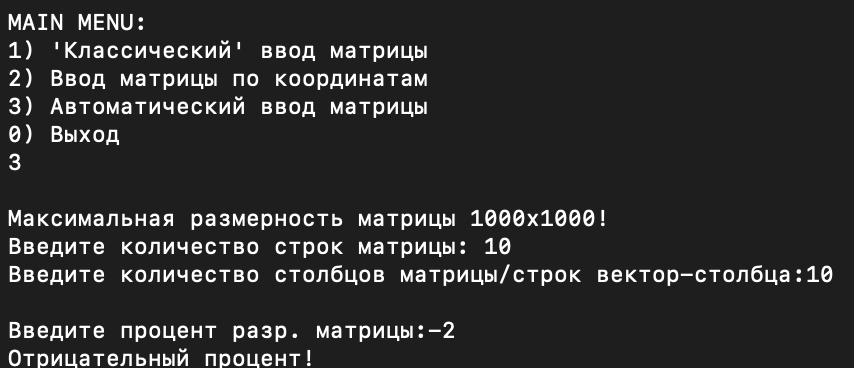
Ввод нуля в строку



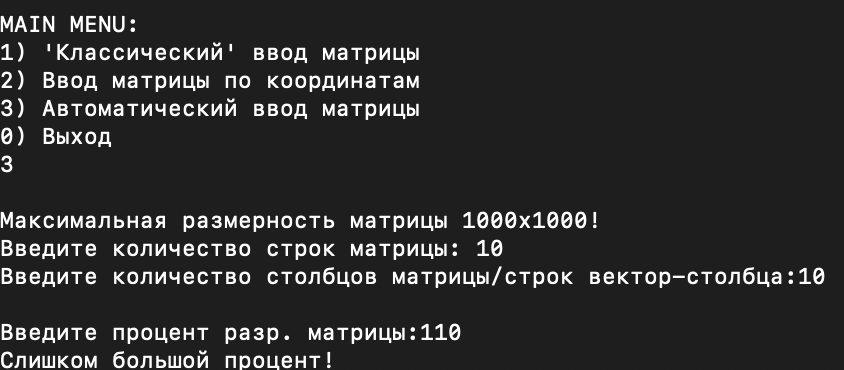
Ввод значения, выходящего за меню



Ввод отрицательного процента



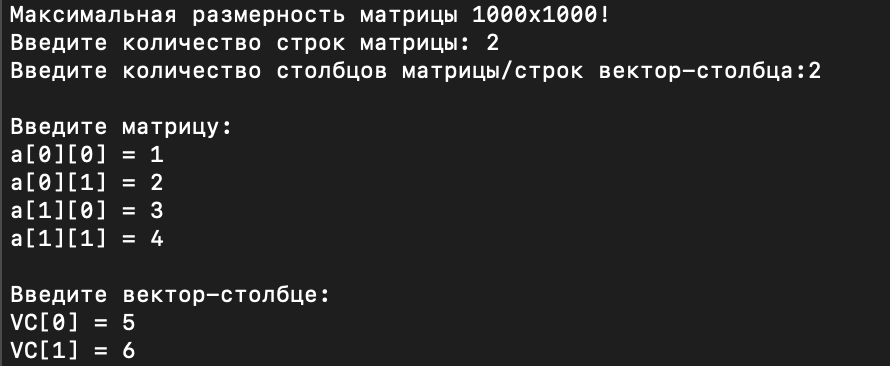
Ввод процента больше 100

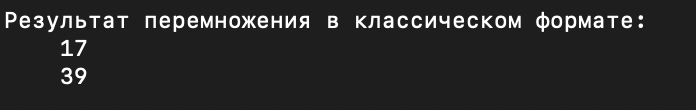


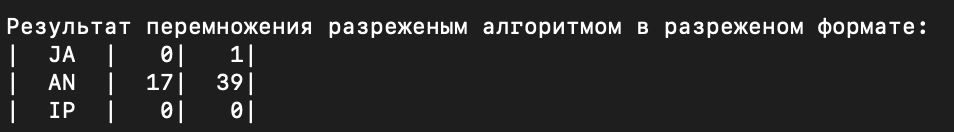
**Тепличные**

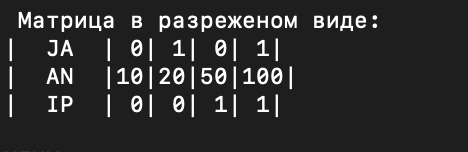
2 матрицы 2х2 со значениями

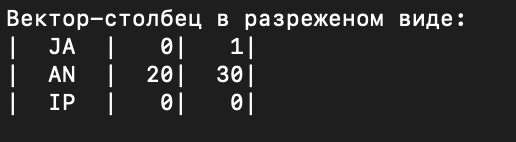
1 2

3 4****

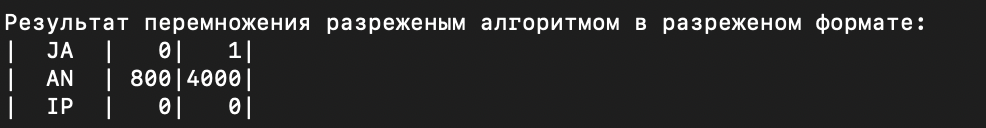
Результат умножения в обычном формате

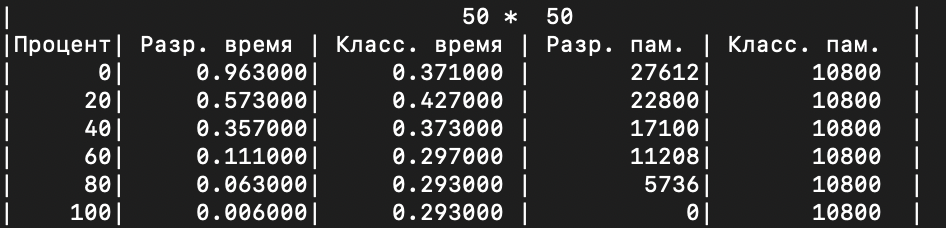
Результат умножения в разреженной форме

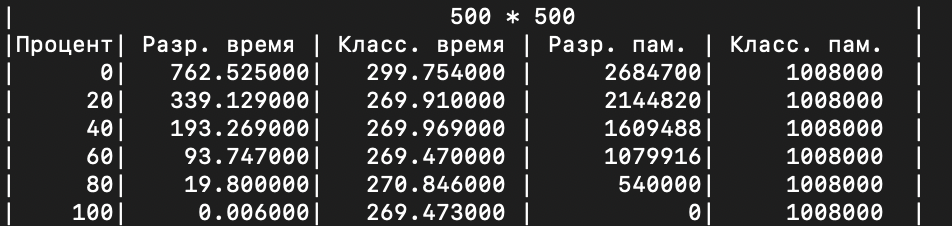
Умножение матриц с координатным вводом:



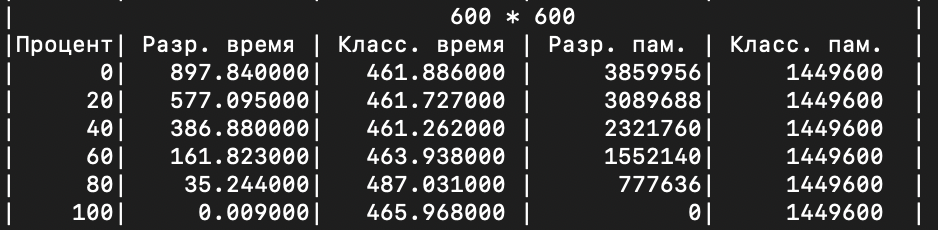
Результат умножения:

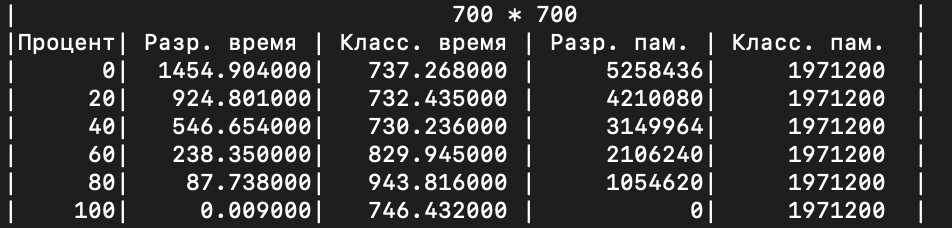


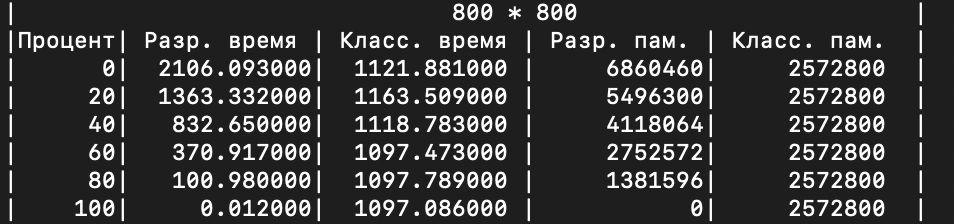
**Сравнение эффективности**



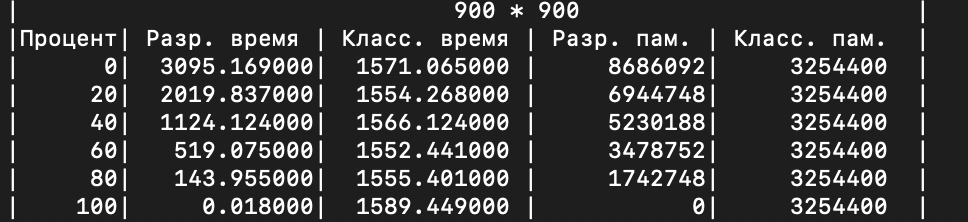
В матрице 500х500 выгода начинается с 30%



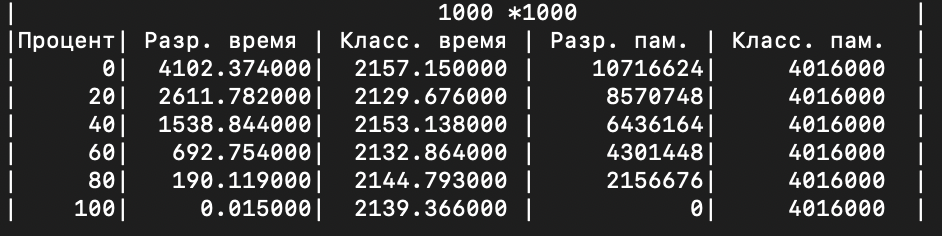
В матрице 600х600 выгода начинается с 30%

В матрице 700х700 выгода начинается с 30%

В матрице 800х800 выгода начинается с 35%



В матрице 900х900 выгода начинается с 35%



В матрице 1000х1000 выгода начинается с 35%

Как видно из полученных таблиц, существует процент разреж. матрицы, при котором алгоритм классического умножения становится более эффективным, чем разреженного. Он примерно равен 30-40 процентам. Заметим, что память, выделяемая на классическую матрицу, постоянна, в то время как для динамической разреженной матрицы она увеличивается в зависимости от количества ненулевых элементов. Таким образом, при заполненности до 60% алгоритм разреженного умножения выигрывает как по памяти, так и по времени, после 60% проигрывает уже по времени, а после 40% - уже и по памяти. Это объясняется тем, что на каждый элемент разреженной матрицы нужно по элементу в двух массивах.

**Ответы на вопросы**

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

Разреженная матрица – матрица, в которой количество ненулевых элементов намного меньше, чем количество нулевых. Существует множество способов хранения таких матриц, к примеру, в виду односвязных линейных списков, стеков, очередей. Самыми популярными способами можно назвать схему Кнута и схему Чанга. Схема Кнута состоит из большого количества массивов, при этом информация, содержащаяся в них, избыточна и дублируется. Данная схема 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). Схема же Чанга и Густавсона, называемая также «разреженным строчным форматом» содержит всего три массива, таким образом, выигрывая по памяти относительно схемы Кнута. Именно данная схема, содержащая массив элементов, массив строк и массив элементов – входов в столбцы и была реализована нами в ходе выполнения лабораторной работы.

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

Под хранение обычной матрицы, как было представлено выше, требуется один и тот же объём памяти, поскольку она состоит из массива массивов фиксированной длины (статические). У разреженных матриц объём вычисляется по-другому: суммарная память, выделяемая под разреженную матрицу, равна сумме всех массивов, использованных для её хранения, а также дополнительных переменных. К примеру, если мы рассмотрим схему Чанга, использованную в нашей лабораторной, то увидим, что объём памяти состоит из двух массивов, количество элементов которых равно количеству ненулевых элементов матрицы, а также массива, в котором количество элементов равно количеству столбцов + 1 (для хранения первого неиспользованного в двух предыдущих массивах индекса).

3. Каков принцип обработки разреженной матрицы?

Разреженная матрица представляет собой набор из трёх массивов, содержащих набор информации, по которой можно восстановить полную матрицу в классическом формате: элемент, номер его строки, а также массив, в котором указаны номера элементов, с которых начинается каждый из столбцов. Таким образом, алгоритм умножения будет выглядеть следующим образом: в целом, это похоже на скалярное умножение, но мы проходим не по всем элементам столбца / строки, а только по ненулевым, а затем по тем, позиции которых были ненулевыми в первом.

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

Используемый алгоритм следует выбирать, исходя из степени разреж. матрицы. Экспериментально было установлено, что граница для времени находится примерно в 30%, для памяти – в районе 40%. То есть, если разреж. ниже, то следует использовать алгоритмы для разреженных матриц, а если выше – то уже классические алгоритмы, в том числе и классическое умножение матриц.

**Выводы по проделанной работе**

В данной лабораторной работе я впервые столкнулся с матрицами, представленными в разреженном виде. Основные трудности были связаны с пониманием обработки матрицы для хранения её в разреженном виде и умножение подобных матриц. Цель лабораторной работы, которую я и смог достичь, - правильный выбор алгоритма хранения матрицы (классический полный или разреженный), в зависимости от степени заполненности / разреженности матрицы, а также, полученный экспериментально, порог, при котором классический формат становится даже выгоднее разреженного.