|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования 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)** |

**Типы и структуры данных.**

**Лабораторная работа №3.**

**«Обработка разреженных матриц»**

Студент **Леонов Владислав Вячеславович**

Группа **ИУ7-36Б**

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Леонов В.В.

*подпись, дата фамилия, и.о.*

Преподаватель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Силантьева А.В.

*подпись, дата фамилия, и.о.*

*2020 г.*

Оглавление

[Условие и техническое задание 3](#_Toc55757144)

[Цель работы 3](#_Toc55757145)

[Задание 3](#_Toc55757146)

[Входные данные 3](#_Toc55757147)

[Выходные данные 4](#_Toc55757148)

[Обращение к программе 4](#_Toc55757149)

[Функции программы 4](#_Toc55757150)

[Аварийные ситуации 4](#_Toc55757151)

[Реализация 5](#_Toc55757152)

[Структуры данных 5](#_Toc55757153)

[Алгоритм 7](#_Toc55757154)

[Тестирование 8](#_Toc55757155)

[Аварийные ситуации 8](#_Toc55757156)

[Штатное поведение 9](#_Toc55757157)

[Оценка эффективности использования стандартного и разреженного вида 12](#_Toc55757158)

[Контрольные вопросы 16](#_Toc55757159)

[1.Что такое разреженная матрица, какие способы хранения вы знаете? 16](#_Toc55757160)

[2. Каким образом и сколько памяти выделяется под хранение разреженной и обычной матрицы? 16](#_Toc55757161)

[3. Каков принцип обработки разреженной матрицы? 16](#_Toc55757162)

[4. В каком случае для матриц эффективнее применять стандартные алгоритмы обработки матриц? От чего это зависит? 16](#_Toc55757163)

[Вывод о проделанной работе 17](#_Toc55757164)

# Условие и техническое задание

## 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.

## Задание

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.

## Входные данные

1. **Матрица и вектор-столбец в стандартном виде**.
2. **Матрица и вектор в разреженном виде**.
3. **Размеры матрицы и вектора, степень их разреженности** для их автоматической генерации.
4. **Опция меню** (целое число от 0 до 9).

## Выходные данные

1. **Матрица, вектор - столбец в стандартом виде.**
2. **Матрица и вектор – столбец в разреженном виде.**
3. **Время выполнения операции умножения**, представленное в виде количества тиков процессора**.**
4. **Сравнительная таблица умножения**, используя стандартный и разреженный метод при разных размерах и степени разреженности матрицы/вектора-столбца**.**

## Обращение к программе

Обращение к программе **app.exe** осуществляется через запуск из терминала.

## Функции программы

1. Ввод матрицы и вектора в стандартом виде.
2. Ввод матрицы и вектора в разреженном виде.
3. Сгенерировать матрицу и вектор автоматически.
4. Умножить матрицу и вектор, используя стандартный вид.
5. Умножить матрицу и вектор, используя разреженный вид.
6. Печать исходных данных и результирующей матрицы в стандартном виде.
7. Печать исходных данных и результирующей матрицы в разреженном виде.
8. Сравнение времени умножения при разном представлении матрицы.
9. Справка.
10. Завершить работу.

## Аварийные ситуации

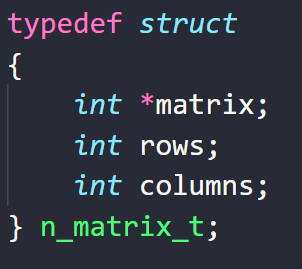
1. Некорректная опция меню (буква или значение, несоответствующее ни одному пункту из меню.
2. Некорректный ввод матрицы или вектора (буква или в случае разряженной матрицы ввод ненулевого значения в одну и ту же ячейку матрицы)
3. Некорректные значения количества столбцов, строк и степени разреженности при генерации данных автоматически.
4. Некорректный размер матрицы и вектора при умножении (количество столбцов матрицы не соответствует количеству строк вектора)
5. Ошибка выделения памяти операционной системой.

# Реализация

## Структуры данных

* Структура **n\_matrix\_t**

Для хранения матрицы и вектора в стандартом виде используется агрегированный тип **n\_matrix\_t.**



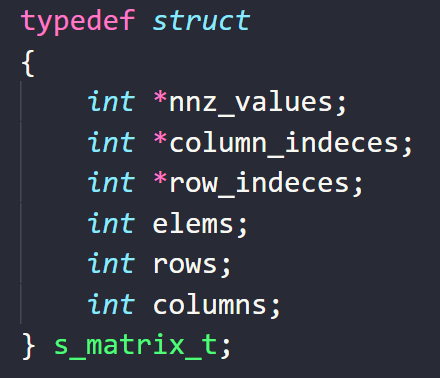
Поле **matrix** представляет собой указатель на динамический массив, элементами которого являются целые числа, и служит для хранения всех элементов матрицы.

Поле **rows** представляет собой целое число и служит для хранения количества строк в матрице.

Поле **columns** представляет собой целое число и служит для хранения количества столбцов в матрице.

* Структура **s\_matrix\_t**

Для хранения матрицы и вектора в разреженном виде используется агрегированный тип **s\_matrix\_t.**

****

Поле **nnz\_values** представляет собой указатель на целочисленный динамический массив, который служит для хранения всех ненулевых значений матрицы.

Поле **column\_indeces** представляет собой указатель на динамический массив, элементами которого являются целые числа, и служит для хранения значений соответствующих столбцов матрицы для ненулевых значений матрицы.

Поле **row\_indeces** представляет собой указатель на динамический массив, элементами которого являются целые числа, и служит для хранения индексов начала строки матрицы среди ненулевых значения (i-элемент массива соответствует i-строке матрицы). В случае, если i-строка пустая, то в соответствующую ячейку записывается значение **-1.**

Поле **elems** представляет собой целое число и служит для хранения количества ненулевых значений матрицы.

Поле **rows** представляет собой целое число и служит для хранения количества строк в матрице.

Поле **columns** представляет собой целое число и служит для хранения количества столбцов в матрице.

## Алгоритм

1. Пользователь вводит номер команды из меню.
2. Пока пользователь не введет 0 (выход из программы), ему будет предложено выполнять действия из меню программы.

Для реализации чтения матрицы и вектора в стандартном виде используются функции **read\_nmatrix** и **read\_nvector** соответственно. Выполняется запрос размерностей с последующим ввод самих элементов. После выполнения чтения матрицы в стандартом виде выполняется автоматическая генерация разреженной матрицы, соответствующей введенной (функция **form\_sparce**).

Для реализации чтения матрицы и вектора в разреженном виде используются функции **read\_smatrix** и **read\_svector** соответственно. Выполняется запрос размерностей, количество ненулевых элементов и заполнение соответствующих разреженной матрице динамических массивов. После выполнения чтения матрицы в разреженном виде выполняется автоматическая генерация стандартной матрицы, соответствующей введенной (функция **form\_normal**).

Для реализации автоматического создания матриц используется функция **generator.** Выполняется запрос размерностей и степени разреженности в процентах, с последующим созданием и заполнением случайными числами матрицы в стандартном виде. После выполнения генерации матрицы в стандартом виде выполняется автоматическая генерация разреженной матрицы, соответствующей созданной (функция **form\_sparce**).

Для реализации умножения матрицы и вектора в стандартном и разреженном виде используются функции **n\_multi** и **s\_multi** соответственно. Задействуется «обычный» алгоритм умножения строки на столбец.

Для реализации печати матрицы и вектора в стандартом и разреженном виде используются функции **print\_n** и **print\_m**.

# Тестирование

## Аварийные ситуации

|  |  |  |  |
| --- | --- | --- | --- |
| № | Описание теста | Входные данные | Сообщение |
| 1 | Некорректная опция из меню | Введите опцию: abc  Введите опцию:101 | Некорректный ввод |
| 2 | Некорректный ввод матрицы или вектора | Введите количество  строк: ывф  Введите элемент  матрицы: d.e3e | Ошибка чтения |
| 3 | Некорректные значения размерностей и степени разрежённости при автоматической генерации | Введите количество  строк: ыв.32ф  Введите процент разреженности:101 | Некорректный ввод |
| 4 | Некорректный размер матрицы и вектора при умножении | Матрица [5x5]  Вектор [1x10] | Некорректный размер матриц |
| 5 | Ошибка выделения памяти операционной системой | Введите количество строк:  1000000000  Введите количество столбцов:  1000000000 | Ошибка выделения памяти |

## Штатное поведение

|  |  |  |
| --- | --- | --- |
| № | Описание теста | Демонстрация работы программы |
| 1 | Чтение матрицы и вектора в стандартном виде |  |
| 2 | Чтение матрицы и вектора в разреженном виде |  |
| 3 | Сгенерировать матрицу и вектор автоматически |  |
| 4 | Умножение матрицы и вектора в стандартном/разреженном виде (не пустые) |  |
| 5 | Умножение матрицы и вектора в стандартном/разреженном виде (пустые) |  |
| 6 | Печать данных (не пустые) |  |
| 7 | Печать данных (пустые) |  |
| 8 | Сравнение времени умножения при разном представлении матриц |  |
| 9 | Справка |  |
| 10 | Завершение работы |  |

# Оценка эффективности использования стандартного и разреженного вида

Для усреднения оценки времени и снижения погрешности, связанной с исходным состоянием матрицы, замеры были проведены многократно (1000 раз). Таким образом, приведенное ниже время является средним для 1000 операций.

**Матрица 100x100**

Время (в тиках процессора)

|  |  |  |
| --- | --- | --- |
| Степень заполненности | Стандартный вид | Разреженный вид |
| 10% | 129408 | 11390 |
| 20% | 133297 | 19014 |
| 30% | 129250 | 35610 |
| 40% | 131082 | 66297 |
| 50% | 127232 | 110681 |
| 60% | 133878 | 125436 |
| 70% | 127648 | 130421 |
| 80% | 132148 | 128726 |
| 90% | 134017 | 147634 |
| 100% | 133689 | 159871 |

Объем занимаемой памяти (в байтах)

|  |  |  |
| --- | --- | --- |
| Степень заполненности | Стандартный вид | Разреженный вид |
| 10% | 80016 | 16184 |
| 20% | 32824 |
| 30% | 48824 |
| 40% | 64824 |
| 50% | 80824 |
| 60% | 96824 |
| 70% | 112824 |
| 80% | 128824 |
| 90% | 144824 |
| 100% | 160824 |

**Матрица 200x200**

Время (в тиках процессора)

|  |  |  |
| --- | --- | --- |
| Степень заполненности | Стандартный вид | Разреженный вид |
| 10% | 525955 | 38836 |
| 20% | 518945 | 104082 |
| 30% | 509323 | 220453 |
| 40% | 508321 | 392532 |
| 50% | 504325 | 512934 |
| 60% | 507935 | 592322 |
| 70% | 509214 | 630329 |
| 80% | 504435 | 629999 |
| 90% | 509214 | 637890 |
| 100% | 531455 | 637991 |

Объем занимаемой памяти (в байтах)

|  |  |  |
| --- | --- | --- |
| Степень заполненности | Стандартный вид | Разреженный вид |
| 10% | 320016 | 64824 |
| 20% | 128824 |
| 30% | 192824 |
| 40% | 256824 |
| 50% | 320824 |
| 60% | 384824 |
| 70% | 448824 |
| 80% | 512824 |
| 90% | 576824 |
| 100% | 640824 |

**Матрица 400х400**

Время (в тиках процессора)

|  |  |  |
| --- | --- | --- |
| Степень заполненности | Стандартный вид | Разреженный вид |
| 10% | 1990824 | 161311 |
| 20% | 2004265 | 523233 |
| 30% | 2046338 | 1021214 |
| 40% | 2041999 | 1632451 |
| 50% | 2001211 | 2110090 |
| 60% | 2111343 | 2236709 |
| 70% | 2023323 | 2310007 |
| 80% | 2042122 | 2297937 |
| 90% | 1998592 | 2379012 |
| 100% | 2103122 | 2423405 |

Объем занимаемой памяти (в байтах)

|  |  |  |
| --- | --- | --- |
| Степень заполненности | Стандартный вид | Разреженный вид |
| 10% | 1280016 | 259224 |
| 20% | 515224 |
| 30% | 771224 |
| 40% | 1027224 |
| 50% | 1283224 |
| 60% | 1539224 |
| 70% | 1795224 |
| 80% | 2051224 |
| 90% | 2307224 |
| 100% | 2563224 |

**Итоги**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Степень заполненности | 100x100 | | 200x200 | | 400x400 | |
| [1] | [2] | [1] | [2] | [1] | [2] |
| 10% | +91% | +80% | +92% | +79% | +92% | +80% |
| 20% | +85% | +60% | +79% | +59% | +73% | +60% |
| 30% | +72% | +39% | +56% | +40% | +50% | +40% |
| 40% | +49% | +19% | +22% | +20% | +20% | +20% |
| 50% | +13% | -1% | -1% | -1% | -5% | +0% |
| 60% | +6% | -21% | -16% | -20% | -6% | -20% |
| 70% | -2% | -41% | -23% | -40% | -14% | -40% |
| 80% | +2% | -61% | -24% | -60% | -12% | -60% |
| 90% | -10% | -81% | -25% | -81% | -19% | -80% |
| 100% | -20% | -101% | -20% | -100% | -15% | -100% |

[1] Преимущество по времени использования разреженной матрицы вместо обычной (в процентах)

[2] Преимущество по памяти использования разреженной матрицы вместо обычной (в процентах)

# Контрольные вопросы

## 1.Что такое разреженная матрица, какие способы хранения вы знаете?

Разреженная матрица – это матрица, содержащая большое количество нулей. Способы хранения: строчный формат, столбцовый формат, линейный связный список, кольцевой связный список.

## 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 формата, требуется (2 \* K + N) ячеек памяти, где K – количество ненулевых элементов, N – количество строк в разреженной матрице.

## 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 методы обработки матрицы эффективно применять в случае, если степень заполненности матрицы порядка 40~50%. В противном случае более эффективно использовать метод обработки разреженных матриц, это дает выигрыш и по памяти, и по времени выполнения операций.

# Вывод о проделанной работе

В ходе лабораторного практикума мы изучили обработку разреженных матриц, закрепили умения работы с записями и динамической памятью, методы организации пользовательского интерфейса программы.

Выделили преимущества и недостатки представления разреженных матриц. Использование разреженных матриц позволяет существенно сократить количество требуемых ресурсов при заполненности матрицы менее 40~50% (по скорости до 92%, по памяти до 80%). При заполненности матрицы более 40~50% использование разреженных матриц не имеет смысл (наблюдается проигрыш по времени ~20%, до 100% по памяти).