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 ИУ

КАФЕДРА ИУ7

ОТЧЕТ ПО ЛР3 ТИПОВ И СТРУКТУР ДАННЫХ, Вариант 3

Студент Андреев Александр Алексеевич
фамилия, имя, отчество

Группа ИУ7-34Б

Тип практики учебная

Студент _____ _____
подпись, дата фамилия, и.о.

Преподаватель _____ _____
подпись, дата фамилия, и.о.

Оценка _____

9 октября 2020

Оглавление

1. Цель работы

В качестве цели работы ставится реализация алгоритма обработки разреженных матриц, сравнение эффективности использования этих алгоритмов по времени выполнения и требуемой памяти со стандартными алгоритмами обработки матриц при различном процентном заполнении матриц ненулевыми значениями и при различных размерах матриц.

2. Описание условия задачи

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 вектора-строки и матрицы, 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.

3. Описание ТЗ, включающее внешнюю спецификацию

a. Описание исходных данных

Для каждой операции указано, как задавать к ней параметры, а также будет ли требоваться ручной ввод из из консоли (и предложено указать формат - как в системе или классический) или же значения будут сгенерированы системой.

Ограничения на вводимые данные:

- Значения по умолчанию все элементы матриц представлены в виде 4-байтового целого
- Значения для размеров только положительны
- Для процентной разреженности - значения неотрицательны

b. Описание задачи, реализуемой программой

Программа имеет два уровня меню, где первое - Основное, второе - Меню операций.

В основном меню программа должна предложить пользователю на выбор ввод данных через терминал, из текстового файла и посредством генерации.

Рис. 1 (Основное меню программы)

```
Основное меню программы:
  1 - Ручной ввод матрицы
  2 - Чтение матрицы из файла
  3 - Генерация матрицы

  Укажите пункт меню (1-3):
```

После выбора соответствующего пункта Основного меню и ввода матрица Пользователь должен получить возможность обращаться с Меню операций.

Рис. 2 (Меню операций программы)

```
Меню операций:
  1 - Умножение вектора-строки на матрицу, хранящихся в разреженной форме, с получением результата в той же форме
  2 - Умножение стандартным алгоритмом работы с матрицами
  3 - Сравнение времени выполнения операций, объема памяти при использовании этих алгоритмов при различной доли заполненности матриц
  4 - Вывод в разреженной форме
  5 - Вывод в простой форме
  0 - Завершение программы

  Укажите пункт меню (0-5):
```

c. Способ обращения программы

Скомпилированная программа запускается командой **“./a.out X”** на Unix и **“./a.exe X”**.

После запуска пользователем программа должна вывести инструкцию по своему применению, Основное меню и приглашение к выбору пункта меню (Рис. 3).

Рис. 3

```
Программа умножения вектора-строки на матрицу,
хранящихся в разреженной форме, с получением результата в той же форме.

Основное меню программы:
  1 - Ручной ввод матрицы
  2 - Чтение матрицы из файла
  3 - Генерация матрицы
```

```
Укажите пункт меню (1-3):
```

Далее после ввода соответствующей команды программа должна выполнять команды пользователя.

- **Комада “1”**
  При указании пункта меню “1” Пользователь получает возможность ввести матрицу вручную.
- **Команда “2”**
  При указании пункта меню “2” Пользователь получает возможность считать матрицу из файла.
- **Команда** “**3”**
  При указании пункта меню “2” Пользователь получает возможность считать матрицу из файла.

В случае ошибки ввода Пользователя при указании пункта Основного меню Программа должна сообщить об этом.

Далее после ввода соответствующей команды программа должна выполнять команды пользователя.

Рис. 4

```
Меню операций:
  1 - Умножение вектора-строки на матрицу, хранящихся в разреженной форме, с получением результата в той же форме
  2 - Умножение стандартным алгоритмом работы с матрицами
  3 - Сравнение времени выполнения операций, объема памяти при использовании этих алгоритмов при различной доли заполненности матриц
  4 - Вывод в разреженной форме
  5 - Вывод в простой форме
  0 - Завершение программы

  Укажите пункт меню (0-5):
```

Далее после ввода соответствующей команды программа должна выполнять команды пользователя.

- **Команда “0”**
  При указании пункта меню “0” Пользователем программа должна завершить свою работу.

- **Команда “1”**
  При указании пункта меню “1” Пользователем программа должна предложить пользователю ввод матрицы, хранящуюся в разреженной форме, с дальнейшим умножением и выводом.
- **Команда** “**2”**
  При указании пункта меню “1” Пользователем программа должна предложить пользователю ввод матрицы с дальнейшим умножением и выводом стандартным способом.
- **Команда** “**3”**
  При указании пункта меню “3” Пользователем программа должна запросить у пользователя заполненность матрицы и вывести сравнительную информацию по обоим способам.
- **Команда** “**4”**
  При указании пункта меню “1” Пользователем программа должна вывести на экран матрицу в разреженной форме.
- **Команда** “**5”**
  При указании пункта меню “1” Пользователем программа должна вывести на экран матрицу в простой форме.

При наличии аварийных ситуаций и ошибок пользователя программа должна вывести соответствующее сообщение и не должна завершить свою работу абортно.

d. Описание возможных аварийных ситуаций и ошибок пользователя

В представленной ниже **Таблице 1** отражены действия программы при различных возможных, допущенных пользователем, при ее использовании ошибках.

Табл.1

Действие программы при различных ошибках		
№	Действие пользователя	Реакция программы
1	Введен несуществующий номер или строка пункта меню	Программа выведет сообщение об ошибке “Возникла ошибка при вводе. Повторите попытку...”
2	Введено нецелое значение, как элемент матрицы, в ручном режиме или считывании из файла	Программа выведет сообщение об ошибке “Возникла ошибка при вводе. Повторите попытку...”
3	Введен некорректный адрес при заполнении матрицы в ручном режиме или считывании из	Программа выведет сообщение об ошибке

	файла	“Указан неверный размер. Повторите попытку...”
4	Пользователь умножает матрицу стандартным способом, где все элементы нулевые	Программа выведет сообщение об ошибке “В матрице все элементы нулевые”

4. Описание внутренних СД

Считанные из консоли числа внутри программы хранятся в виде структуры **number,** которая представляет из себя (Рис. 1)
INPUT_STRING_MAX_SIZE = 256;
INPUT_TABLE_INFORMATION_MAX_SIZE = 256;

Рис. 5

```
struct ja
{
   int num;
   struct ja *next;
};
```

5. Алгоритм

Основная программа
Считав матрицу и находясь в меню операций программы, программа до получения продолжает свою работу до получения от пользователя указаний о выходе из нее.

Выбор действия внутри меню операций:

- **Команда “0”.** При указании пункта меню “0” Пользователем программа должна завершить свою работу.
- **Команда “1”**. При указании пункта меню “1” Пользователем программа должна предложить пользователю ввод матрицы, хранящуюся в разреженной форме, с дальнейшим умножением и выводом.
- **Команда** “**2”.** При указании пункта меню “1” Пользователем программа должна предложить пользователю ввод матрицы с дальнейшим умножением и выводом стандартным способом.
- **Команда** “**3”.** При указании пункта меню “3” Пользователем программа должна запросить у пользователя заполненность матрицы и вывести сравнительную информацию по обоим способам.
- **Команда** “**4”.** При указании пункта меню “1” Пользователем программа должна вывести на экран матрицу в разреженной форме.

- **Команда** “**5”.** При указании пункта меню “1” Пользователем программа должна вывести на экран матрицу в простой форме.

Описание основных функций:
int mult_std_matrix(int **matrix, int size_rows, int size_cols, int c);
Функция, производящая операцию умножения матрицы, хранящейся в стандартном виде.

int mult_matrix(int *A, int *IA, int*JA, int count, int size, int c);
Функция умножения матрицы, хранящейся в разреженном виде.

int ** manual_input(int *size_rows, int *size_cols, int *count);
Функция ручного ввода матрицы. Здесь пользователь вводит сначала размерность матрицы, после чего указывает количество ненулевых элементов, далее по позициям вводит значения ненулевых элементов

int **input_from_file(int *size_cols, int *size_rows, int *count);
Чтение матрицы из файла.

void output_std_matrix(int **mtr, int size_rows, int size_cols);
Вывод матрицы, хранящейся в стандартном виде, на экран.

void output_matrix(int *A, int *IA, int*JA, int count, int size_cols);
Вывод матрицы, хранящейся в разреженном виде, на экран.

void add_elem(struct ja **JA)
Добавление элементов в матрицу, хранящуюся в разреженном виде.

6. Набор тестов с указанием, что проверяется

В представленных ниже **Таблица 2** отражены тестирование устойчивости работы консольного меню программы и демонстрация устойчивости работы программы к различному типу выполняемых с ней операций пользователем соответственно.

Табл. 2

Тестирование устойчивости работы программы			
№	Ввод пользователя	Реакция программы	Что проверяется данной операцией?
1	При вводе пользователем неподходящего пункта меню.	Программа выводит “Возникла ошибка при вводе. Повторите попытку...”	Устойчивость программы к вводу неверной команды меню
2	Заполнение элементов матрицы не целыми числами.	Программа выводит “Возникла ошибка при вводе.	Устойчивость программы к вводу неверной команды внутри пунктов

		Повторите попытку...”	меню
3	Пользователем введен несуществующий или пустой файл	Программа выводит “Допущена ошибка при открытии файла. Повторите попытку...”	Устойчивость к открытию несуществующего или пустого файла
4	Укажите пункт меню (1-3): 1 Размерность матрицы через пробел: 4 4 Количество ненулевых элементов: 2 Номер строки: 0 Номер столбца: 0 Укажите значение:1 Номер строки: 1 Номер столбца: 0 Укажите значение:40	Укажите пункт меню (0-5): 5 1 0 0 0 40 0 0 0 0 0 0 0 0 0 0 0	Корректный вывод матрицы в стандартной форме
5	Укажите пункт меню (1-3): 1 Размерность матрицы через пробел: 4 4 Количество ненулевых элементов: 2 Номер строки: 0 Номер столбца: 0 Укажите значение:1 Номер строки: 1 Номер столбца: 0	Укажите пункт меню (0-5): 4 A: 1 40 IA: 0 1 JA: 0 2 2 2 2	Вывод матрицы в разреженной форме

	Укажите значение:40		
6	Укажите пункт меню (1-3): 3 Введите размер матрицы и процент заполнения: Высота = 5 Ширина = 5 Заполненность = 50 Матрица генерируется... Меню операций: 1 - Умножение вектора-строки на матрицу, хранящихся в разреженной форме, с получением результата в той же форме 2 - Умножение стандартным алгоритмом работы с матрицами 3 - Сравнение времени выполнения операций, объема памяти при использовании этих алгоритмов при различной доли заполненности матриц 4 - Вывод в разреженной форме 5 - Вывод в простой форме 0 - Завершение программы Укажите пункт меню (0-5): 5	262 17 0 0 372 0 0 0 0 0 635 101 0 829 624 0 931 352 0 428 0 535 0 0 168	Генерация матрицы
7	Укажите пункт меню (1-3): 3 Введите размер матрицы и процент заполнения: Высота = 10000 Ширина = 10000 Заполненность = 60 Матрица генерируется...	Время умножения матрицы на вектор в стандартном виде: 897260242 тиков Память умножения матрицы на вектор в стандартном виде: 400000000 байт Время умножения матрицы на вектор в разреженом виде: 995532466 тиков Память умножения	Вывод сравнительной информации

	Меню операций: 1 - Умножение вектора-строки на матрицу, хранящихся в разреженной форме, с получением результата в той же форме 2 - Умножение стандартным алгоритмом работы с матрицами 3 - Сравнение времени выполнения операций, объема памяти при использовании этих алгоритмов при различной доли заполненности матриц 4 - Вывод в разреженной форме 5 - Вывод в простой форме 0 - Завершение программы Укажите пункт меню (0-5): 6	матрицы на вектор в разреженом виде: 136442616 байт	

7. Выводы по проделанной работе

Табл. 3

Сравнение работы алгоритмов					
Размерность	Заполненность, %	Стандартная		Разреженная	
		Время	Память	Время	Память
200 x 200	0	0.000428 сек	160000 байт	0.000355 сек	800 байт
	25	0.000146 сек		0.000194	80800 байт
	50	0.000167 сек		0.000365 сек	160800 байт
	75	0.000199 сек		0.000285 сек	240800 байт
	85	0.000168 сек		0.000242 сек	272800 байт
	95	0.000184 сек		0.000188 сек	304800 байт
	100	0.000191 сек		0.000167 сек	320800 байт
500 x 500	0	0.002425 сек	1000000 байт	0.000571 сек	2000 байт
	25	0.001844 сек		0.001338 сек	502000 байт

	50	0.001378 сек		0.001850 сек	1002000 байт
	75	0.001628 сек		0.001468 сек	1502000 байт
	85	0.001196 сек		0.001025 сек	1702000 байт
	95	0.001710 сек		0.001038 сек	1902000 байт
	100	0.001254 сек		0.000886 сек	2002000 байт
1000 x 1000	0	0.008937 сек	4000000 байт	0.001105 сек	4000 байт
	25	0.006117 сек		0.004243 сек	2004000 байт
	50	0.006729 сек		0.006501 сек	4004000 байт
	75	0.006124 сек		0.005330 сек	6004000 байт
	85	0.006953 сек		0.004279 сек	6804000 байт
	95	0.006520 сек		0.003619 сек	7604000 байт
	100	0.008567 сек		0.002762 сек	8004000 байт
2000 x 2000	0	0.060414 сек	16000000 байт	0.002220 сек	8000 байт
	25	0.052559 сек		0.016659 сек	8008000 байт
	50	0.028169 сек		0.025930 сек	16008000 байт
	75	0.058828 сек		0.020201 сек	24008000 байт
	85	0.028452 сек		0.016434 сек	27208000 байт
	95	0.058405 сек		0.012463 сек	30408000 байт
	100	0.031450 сек		0.011917 сек	32008000 байт
3000 x 3000	0	0.136901 сек	36000000 байт	0.003236 сек	12000 байт
	25	0.122650 сек		0.038026 сек	18012000 байт
	50	0.119400 сек		0.058392 сек	36012000 байт
	75	0.121427 сек		0.045747 сек	54012000 байт

	85	0.121012 сек		0.037446 сек	61212000 байт
	95	0.121303 сек		0.028208 сек	68412000 байт
	100	0.118081 сек		0.024641 сек	72012000 байт

Полученные результаты позволяют сделать следующие выводы:

1) Разреженный вид становится эффективней по памяти при разреженности более чем 50-60%;

2) Разреженный вид становится эффективней по времени при разреженности более чем 75-80%;

3) При разреженности меньше 40% разреженный вид гораздо менее эффективный, нежели классический;

4) Чем больше матрица, тем сильней проявляется эффективность разряженного метода хранения (при соответствующих разреженностях).

Таким образом:

1) Программист не должен использовать разреженный способ хранения матриц при разреженности меньше 40%.

2) Программист должен использовать разреженный способ хранения матриц при разреженности больше 70%.

3) В остальных случаях выбор метода решения остаётся за программистом и зависит от доступных ресурсов и поставленных целей.

Использование разреженной матрицы оправдано при большой размерности и небольшом заполнении. В таком случае можно получить значительный выигрыш по памяти и по времени. Однако в этом случае значительно усложняется алгоритм обработки матриц.

Время выполнения стандартного алгоритма почти линейно зависит от размерности матрицы. Этот алгоритм эффективен при высоком заполнении матрицы. Однако при заполнении матрицы менее 20-15% разреженный алгоритм позволяет добиться более высокой скорости работы при использовании меньшего количества памяти.

Сложность разреженного алгоритма связана с невозможностью доступа к элементам по индексам.

При заполнении матрицы более, чем на 15-20%, и размерности менее 50*50, безусловно, целесообразнее использовать стандартные способы обработки и хранения матриц.

8. Контрольные вопросы

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

Разреженная матрица – матрица с преимущественно нулевыми элементами.
Число ненулевых элементов в матрице порядка n может выражаться как n^(1+g), где g < 1. Значения g лежат в интервале 0.2 ... 0.5.

Существуют различные методы хранения элементов матрицы в памяти.

Например, линейный связный список, т.е. последовательность ячеек, связанных в определенном порядке. Каждая ячейка списка содержит элемент списка и указатель на положение следующей ячейки.

Можно хранить матрицу, используя кольцевой связный список, двунаправленные стеки и очереди.

Существует диагональная схема хранения симметричных матриц, а также связные схемы разреженного хранения.

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).

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

Для хранения обычной матрицы необходимо: N * M * sizeof(elem) памяти.

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

3. Каков принцип обработки разреженной матрицы?

Обработка разреженной матрицы предполагает работу только с ненулевыми элементами (таким образом, количество операций пропорционально количеству ненулевых элементов).

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

Нам следует учитывать разреженность матрицы, если из этого можно извлечь выгоду за счёт игнорирования нулевых элементов.

Нужно заметить, что происходит падение эффективности по времени ри достижении определенного процента заполненности матрицы ненулевыми элементами.