



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)

ФАКУЛЬТЕТ «ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ»

КАФЕДРА «ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭВМ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3 ПО ДИСЦИПЛИНЕ: ТИПЫ И СТРУКТУРЫ ДАННЫХ

Обработка разреженных матриц

Вариант № 5

Студент **Федин А.А.**

Группа **ИУ7-34Б**

Название предприятия **НУК ИУ МГТУ им. Н. Э. Баумана**

Студент _____ **Федин А.А.**

Преподаватель _____ **Никульшина Т.А.**

Описание условий задачи

Разработать программу умножения разреженных и обычных 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.

Разреженные матрицы хранятся в двух форматах:

Первая разреженная (содержащая много нулей) матрица хранится в форме 3-х объектов: - вектор A содержит значения ненулевых элементов; - вектор IA содержит номера строк для элементов вектора A; - вектор JA, в элементе N_k которого находится номер компонент в A и IA, с которых начинается описание столбца N_k матрицы A.

Вторая разреженная матрица хранится в форме 3-х объектов: - вектор B содержит значения ненулевых элементов; - вектор JB содержит номера столбцов для элементов вектора A; - вектор IB, в элементе N_k которого находится номер компонент в B и JB, с которых начинается описание строки N_k матрицы B.

1. Смоделировать операцию умножения двух матриц, хранящихся в указанной форме, с получением результата в форме хранения первой матрицы.
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

Описание входных данных

На вход программе передаются матрицы.

Матрицы могут содержаться в двух форматах: - Обычный - в начале передается количество строк и столбцов в беззнаковом целочисленном формате, количество строк и столбцов должно быть больше нуля. Далее построчно вводятся сами элементы матрицы, их количество должно соответствовать заявленному. - Координатный - в начале передается количество строк, количество столбцов и количество ненулевых элементов в матрице. Количество строк и столбцов должно быть больше нуля, количество ненулевых элементов не должно превышать количества элементов в матрице. Далее вводятся строки в формате “значение_элемента номер_строки номер_столбца”, номер строки и столбцов элемента должно быть валидно.

Матрицы могут быть получены либо из файлов либо из консоли. Буфер ввода пути файла ограничен 200 символами, в конце файла должен быть переход на новую строку. Программа должна иметь доступ к чтению этого файла.

Формат ввода разреженной матрицы ничем не отличается от формата вывода обычной матрицы

Описание результирующих данных

Результат выполнения программы зависит от выбранной опции вывода данных и способа перемножения матриц:

- 1) Обычная матрица
 - Обычный способ вывода: На каждой строке файла/консоли выводится значение соответствующей строки матрицы.
 - Координатный формат: для каждого элемента в файл/консоль выводится строка в формате “значение_элемента номер_строки номер_столбца”
- 2) Разреженная матрица A
 - Обычный способ вывода: В файл вывода /консоль выводится векторы A, IA, JA со значениями соответствующих элементов
 - Координатный формат: для каждого элемента в файл выводится строка в формате “значение_элемента номер_строки номер_столбца”
- 3) Разреженная матрица B
 - Обычный способ вывода: В файл вывода /консоль выводится векторы B, IB, JB со значениями соответствующих элементов
 - Координатный формат: для каждого элемента в файл выводится строка в формате “значение_элемента номер_строки номер_столбца”

Описание задачи, реализуемой в программе

Создание интерфейса для работы с разреженными и обычными матрицами, сравнение эффективности перемножения обычных и разреженных матриц по времени и памяти для различных размеров матрицы и процентного заполнения.

Описание способов обращения к программе

Способ обращения к программе пользователем происходит через исполняемый файл app.exe.

Описание аварийных ситуаций

- ERROR_FILE - ошибка ввода/вывода, программа не может получить доступ к файлу или файла не существует
- ERROR_INPUT_SIZE - ошибка чтения размеров, программа не смогла прочитать размеры матрицы из файла
- ERROR_INPUT_SIZE_VAL - ошибка значения размеров, размеры матрицы не соответствуют требуемому формату
- ERROR_INPUT - ошибка чтения элемента, программа не смогла прочитать размеры матрицы из файла
- ERROR_INDEX - ошибка добавления элемента по координате, значение строки/столбца элемента превышает допустимый для конкретной матрицы
- ERROR_MALLOC - ошибка выделения динамической памяти, программа не смогла выделить динамическую память на матрицу

- ERROR_WRONG_MATR_SIZE - ошибка размера матрицы, невозможно перемножить матрицы из-за несовместимости размеров матрицы
- ERROR_STR_INPUT - ошибка ввода пути файла, программа не смогла считать путь к файлу с данными
- ERROR_STR_SIZE - ошибка размера строки, путь к требуемому файлу превышает размер буфера
- Другие ошибки и исключительные ситуации, связанные с операционной системой или другими программами, которые пользователь мог вызывать до запуска данной программы.

Описание внутренних СД

В программе для хранения матриц используются три структурных типа данных:

1)full_matrix_t - структурная переменная для хранения обычной матрицы

```
typedef struct
{
    size_t n;
    size_t m;
    double **data;
}full_matrix_t;
```

- **data - матрица значений, формат чисел в матрице - double, память выделяется динамически
- n - количество строк матрицы, формат - целый беззнаковый size_t
- m - количество столбцов матрицы, формат - целый беззнаковый size_t

2)sparse_matrix_A_t - разреженная матрица, хранящаяся в формате A

```
typedef struct
{
    double *A;
    int *IA;
    int *JA;
    size_t elem_cnt;
    size_t JA_len;
}sparse_matrix_A_t;
```

- *A - значения ненулевых элементов, длина равна количеству ненулевых элементов, формат чисел в векторе - double, память выделяется динамически
- *IA - номера строк для элементов вектора A, длина равна количеству ненулевых элементов, формат чисел в векторе - int, память выделяется динамически
- *JA - вектор, в элементе Nk которого находится номер компонент в A и IA, с которых начинается описание столбца Nk матрицы A, длина равна количеству столбцов, формат чисел в векторе - int, память выделяется динамически
- elem_cnt - количество элементов в матрице, формат - целый беззнаковый size_t
- JA_len - количество столбцов в матрице, формат - целый беззнаковый size_t

3)sparse_matrix_B_t - разреженная матрица, хранящаяся в формате B

```
typedef struct
{
```

```
double *B;
int *JB;
int *IB;
size_t elem_cnt;
size_t IB_len;
}sparse_matrix_B_t;
```

- *B - значения ненулевых элементов, длина равна количеству ненулевых элементов, формат чисел в векторе - double, память выделяется динамически
- *JB - номера столбцов для элементов вектора B, длина равна количеству ненулевых элементов, формат чисел в векторе - int, память выделяется динамически
- *IB - вектор, в элементе Nk которого находится номер компонент в B и JB, с которых начинается описание строки Nk матрицы B, длина равна количеству строк, формат чисел в векторе - int, память выделяется динамически
- elem_cnt - количество элементов в матрице, формат - целый беззнаковый size_t
- IB_len - количество столбцов в матрице, формат - целый беззнаковый size_t

Описание алгоритмов

Общий алгоритм данных

Начинается выполнение программы в функции main. Пользователю предлагается выбрать дальнейшее действие:

- 1. Ввести первую обычную матрицу
- 2. Ввести вторую обычную матрицу
- 3. Вывести первую обычную матрицу
- 4. Вывести вторую обычную матрицу
- 5. Ввести первую разреженную матрицу
- 6. Ввести вторую разреженную матрицу
- 7. Вывести первую разреженную матрицу
- 8. Вывести вторую разреженную матрицу
- 9. Произвести умножение обычных матриц
- 10. Произвести умножение разреженных матриц
- 11. Произвести измерение
- 0. Завершение работы

В случае ошибки ввода опции, пользователю предостановится ввести опцию заново. В опциях 1 - 10 предлагается выбрать формат матрицы(стандартная / координатная) и способ ввода/вывода(консоль/файл). Для вывода соответствующей матрицы требуется успешное выполнение опции ввода соответствующей матрицы Для умножения требуется успешное выполнение операции ввода перемножаемых матриц При выборе опции "0" программа завершает свою работу

Ввод обычной матрицы в стандартном формате

- На вход подается путь к файлу, содержащий матрицу и указатель на структурный тип данных, куда необходимо записать матрицу
- Возвращается 0, если чтение произошло успешно или код ошибки
- Из файла считываются размеры матрицы и проверяются их значения
- Осуществляется динамическое выделение памяти на структуру, проверяется корректность выделения
- Далее в цикле считываются элементы матрицы, в случае ошибки - память освобождается и файл закрывается

Ввод обычной матрицы в координатном формате

- На вход подается путь к файлу, содержащий матрицу и указатель на структурный тип данных, куда необходимо записать матрицу
- Возвращается 0, если чтение произошло успешно или код ошибки
- Из файла считываются размеры матрицы, количество ненулевых элементов и проверяются их значения
- Осуществляется динамическое выделение памяти на структуру, проверяется корректность выделения
- Далее в цикле считываются элементы матрицы, в случае ошибки - память освобождается и файл закрывается
- При считывании элементов матрицы проверяется корректность их координат

Ввод разреженной матрицы в стандартном формате

- Осуществляется ввод обычной матрицы в стандартном формате
- На вход подается плотная матрица, которую необходимо преобразовать в разреженную и указатель на разреженную матрицу
- Возвращается 0, если чтение произошло успешно или код ошибки
- Осуществляется динамическое выделение памяти на массивы A,JA,JA(или B,IB,JB соответственно), проверяется корректность выделения
- Осуществляется проход по столбцам и строкам матрицы(по строкам и столбцам для матрицы B)
- Если по модулю элемент больше $1e-8$, записываем его значение и один из индексов, увеличиваем счетчик записанных элементов на 1
- Если счетчик записанных элементов на равен нулю, записываем его значение в массив JA(или IB)

Ввод разреженной матрицы в координатном формате

- Осуществляется ввод обычной матрицы в координатном формате
- На вход подается плотная матрица, которую необходимо преобразовать в разреженную и указатель на разреженную матрицу
- Возвращается 0, если чтение произошло успешно или код ошибки
- Осуществляется динамическое выделение памяти на массивы A,JA,JA(или B,IB,JB соответственно), проверяется корректность выделения

- Осуществляется проход по столбцам и строкам матрицы(по строкам и столбцам для матрицы B)
- Если по модулю элемент больше $1e-8$, записываем его значение и один из индексов, увеличиваем счетчик записанных элементов на 1
- Если счетчик записанных элементов на равен нулю, записываем его значение в массив JA(или IB)

Умножение обычных матриц

- На вход подаются две матрицы и указатель на матрицу, куда необходимо записать результат
- Возвращается 0, если чтение произошло успешно или код ошибки
- Проверяется равенство количества строк первой и количества столбцов второй
- Задаются размеры и память под результирующую матрицу
- Осуществляется проход по столбцам и строкам матрицы результата
- Осуществляется проход от нуля до количества строк первой матрицы и суммируются значения в этих строках/столбцах

Умножение разреженных матриц

- На вход подаются две матрицы и указатель на матрицу, куда необходимо записать результат
- Возвращается 0, если чтение произошло успешно или код ошибки
- Проверяется равенство количества строк первой и количества столбцов второй
- Задаются размеры и память под результирующую матрицу
- Если значения произведения элементов больше нуля - значения записываются в результирующую матрицу

Тестирование

Положительные тесты

Описание тестов	Размер первой матрицы	Элементы первой матрицы	Размер второй матрицы	Элементы второй матрицы	Размер результирующей матрицы	Элементы результирующей матрицы
Умножение матрицы 3x3 на матрицу 3x3	3 3	1 2 3 4 5 6 7 8 9	3 3	1 2 3 4 5 6 7 8 9	3 3	30 36 42 66 81 96 102 126 150
Умножение матрицы 3x3 на матрицу 3x3, первая матрица задана в координатном формате	3 3	1 2 3 4 5 6 7 8 9	3 3	1 2 3 4 5 6 7 8 9	3 3	30 36 42 66 81 96 102 126 150
Умножение матрицы 1x3 на матрицу 3x1	1 3	1 2 3	3 1	4 5 6	1 1	32
Умножение разреженных матриц 3x3 на 3x3	3 3	1 2 3 4 5 6 7 8 9	3 3	1 2 3 4 5 6 7 8 9		A: 30 66. 102 36 81 126 42 96 150 IA: 0 1 2 0 1 2 0 1 2 JA: 0 3 6 9
Умножение матрицы 3x3 на матрицу 3x3, первая матрица	3 3	1 2 3 4 5 6 7 8 9	3 3	1 2 3 4 5 6 7 8 9		A: 30 66 102 36 81 126 42 96 150

Описание тестов	Размер первой матрицы	Элементы первой матрицы	Размер второй матрицы	Элементы второй матрицы	Размер результирующей матрицы	Элементы результирующей матрицы
задана в координатном формате						IA: 0 1 2 0 1 2 0 1 2 JA: 0 3 6 9
Вывод матрицы на экран после ввода в стандартном формате						
Тестирование эффективности матриц	-	-	-	-	-	-

Негативные тесты

Описание тестов	Входные данные	Ожидаемый код ошибки
На вход матрице подается несуществующий файл	This_is_not_existing_file	ERROR_FILE
При вводе матрицы передана строка	hello	ERROR_INPUT_SIZE
При вводе матрицы передана пустая строка		ERROR_INPUT_SIZE
При вводе матрицы переданы нулевые размеры	size(0,0)	ERROR_INPUT_SIZE_VAL
При вводе матрицы задаются отрицательные размеры	size(-1,-1)	ERROR_INPUT_SIZE_VAL
При вводе матрицы в обычном формате значений меньше необходимого	size(3,3) [[1,1,1],[2,2,2]][3]	ERROR_INPUT
При вводе матрицы в координатном формате количество ненулевых элементов больше размера матрицы	size(3,3,10000)	ERROR_INPUT_SIZE_VAL
При вводе матрицы в координатном формате координата элемента выходит за пределы матрицы	size(3,3,1) point: 1 100 100	ERROR_INDEX
Попытка умножения матрицы с неправильными размерами	first: size(1,1) second: size(3,3)	ERROR_WRONG_MATR_SIZE

Оценка эффективности

Матрица 10x10

Процент заполнения	Объем памяти (разреженная_1)	Объем памяти (стандартная_1)	Объем памяти (разреженная_2)	Объем памяти (стандартная_2)	Время (разреженная)	Время (стандартная)
10%	164 байт	800 байт	164 байт	800 байт	5000.0 нс	12000.0 нс
20%	284 байт	800 байт	284 байт	800 байт	8000.0 нс	12000.0 нс
30%	404 байт	800 байт	404 байт	800 байт	12000.0 нс	12000.0 нс
40%	524 байт	800 байт	524 байт	800 байт	19000.0 нс	12000.0 нс
50%	644 байт	800 байт	644 байт	800 байт	18000.0 нс	12000.0 нс
60%	764 байт	800 байт	764 байт	800 байт	22000.0 нс	12000.0 нс
70%	884 байт	800 байт	884 байт	800 байт	23000.0 нс	12000.0 нс
80%	1004 байт	800 байт	1004 байт	800 байт	23000.0 нс	12000.0 нс
90%	1124 байт	800 байт	1124 байт	800 байт	22000.0 нс	12000.0 нс
100%	1244 байт	800 байт	1244 байт	800 байт	26000.0 нс	12000.0 нс

Матрица 25x25

Процент заполнения	Объем памяти (разреженная_1)	Объем памяти (стандартная_1)	Объем памяти (разреженная_2)	Объем памяти (стандартная_2)	Время (разреженная_я)	Время (стандартная_я)
10%	848 байт	5000 байт	848 байт	5000 байт	58000.0 нс	183000.0 нс
20%	1604 байт	5000 байт	1604 байт	5000 байт	118000.0 нс	182000.0 нс
30%	2348 байт	5000 байт	2348 байт	5000 байт	171000.0 нс	187000.0 нс
40%	3104 байт	5000 байт	3104 байт	5000 байт	220000.0 нс	199000.0 нс
50%	3848 байт	5000 байт	3848 байт	5000 байт	253000.0 нс	188000.0 нс
60%	4604 байт	5000 байт	4604 байт	5000 байт	317000.0 нс	187000.0 нс
70%	5348 байт	5000 байт	5348 байт	5000 байт	325000.0 нс	182000.0 нс
80%	6104 байт	5000 байт	6104 байт	5000 байт	320000.0 нс	182000.0 нс
90%	6848 байт	5000 байт	6848 байт	5000 байт	317000.0 нс	182000.0 нс
100%	7604 байт	5000 байт	7604 байт	5000 байт	349000.0 нс	181000.0 нс

Матрица 50x50

Процент заполнения	Объем памяти (разреженная_1)	Объем памяти (стандартная_1)	Объем памяти (разреженная_2)	Объем памяти (стандартная_2)	Время (разреженная_я)	Время (стандартная_я)
10%	3204 байт	20000 байт	3204 байт	20000 байт	495000.0 нс	2470000.0 нс
20%	6204 байт	20000 байт	6204 байт	20000 байт	854000.0 нс	1380000.0 нс
30%	9204 байт	20000 байт	9204 байт	20000 байт	1039000.0 нс	1167000.0 нс
40%	12204 байт	20000 байт	12204 байт	20000 байт	1999000.0 нс	1276000.0 нс
50%	15204 байт	20000 байт	15204 байт	20000 байт	2312000.0 нс	1237000.0 нс
60%	18204 байт	20000 байт	18204 байт	20000 байт	2051000.0 нс	1073000.0 нс
70%	21204 байт	20000 байт	21204 байт	20000 байт	2004000.0 нс	1006000.0 нс
80%	24204 байт	20000 байт	24204 байт	20000 байт	1609000.0 нс	848000.0 нс
90%	27204 байт	20000 байт	27204 байт	20000 байт	1532000.0 нс	947000.0 нс
100%	30204 байт	20000 байт	30204 байт	20000 байт	1115000.0 нс	766000.0 нс

Матрица 100x100

Процент заполнения	Объем памяти (разреженная_1)	Объем памяти (стандартная_1)	Объем памяти (разреженная_2)	Объем памяти (стандартная_2)	Время (разреженная_я)	Время (стандартная_я)
10%	12404 байт	80000 байт	12404 байт	80000 байт	3929000.0 нс	10735000.0 нс
20%	24404 байт	80000 байт	24404 байт	80000 байт	5570000.0 нс	8404000.0 нс
30%	36404 байт	80000 байт	36404 байт	80000 байт	6523000.0 нс	6068000.0 нс
40%	48404 байт	80000 байт	48404 байт	80000 байт	7269000.0 нс	5326000.0 нс

Процент заполнения	Объем памяти (разреженная_1)	Объем памяти (стандартная_1)	Объем памяти (разреженная_2)	Объем памяти (стандартная_2)	Время (разреженная)	Время (стандартная)
50%	60404 байт	80000 байт	60404 байт	80000 байт	7682000.0 нс	4642000.0 нс
60%	72404 байт	80000 байт	72404 байт	80000 байт	7710000.0 нс	4096000.0 нс
70%	84404 байт	80000 байт	84404 байт	80000 байт	7276000.0 нс	3750000.0 нс
80%	96404 байт	80000 байт	96404 байт	80000 байт	6479000.0 нс	3575000.0 нс
90%	108404 байт	80000 байт	108404 байт	80000 байт	5527000.0 нс	3569000.0 нс
100%	120404 байт	80000 байт	120404 байт	80000 байт	5230000.0 нс	3448000.0 нс

Матрица 250x250

Процент заполнения	Объем памяти (разреженная_1)	Объем памяти (стандартная_1)	Объем памяти (разреженная_2)	Объем памяти (стандартная_2)	Время (разреженная)	Время (стандартная)
10%	76004 байт	500000 байт	76004 байт	500000 байт	21074000.0 нс	86127000.0 нс
20%	151004 байт	500000 байт	151004 байт	500000 байт	40792000.0 нс	54470000.0 нс
30%	226004 байт	500000 байт	226004 байт	500000 байт	61419000.0 нс	54895000.0 нс
40%	301004 байт	500000 байт	301004 байт	500000 байт	79379000.0 нс	54466000.0 нс
50%	376004 байт	500000 байт	376004 байт	500000 байт	97944000.0 нс	54821000.0 нс
60%	451004 байт	500000 байт	451004 байт	500000 байт	108504000.0 нс	55775000.0 нс
70%	526004 байт	500000 байт	526004 байт	500000 байт	108204000.0 нс	55692000.0 нс
80%	601004 байт	500000 байт	601004 байт	500000 байт	97625000.0 нс	56020000.0 нс
90%	676004 байт	500000 байт	676004 байт	500000 байт	87340000.0 нс	56012000.0 нс
100%	751004 байт	500000 байт	751004 байт	500000 байт	77004000.0 нс	54563000.0 нс

Матрица 500x500

Процент заполнения	Объем памяти (разреженная_1)	Объем памяти (стандартная_1)	Объем памяти (разреженная_2)	Объем памяти (стандартная_2)	Время (разреженная)	Время (стандартная)
10%	302004 байт	2000000 байт	302004 байт	2000000 байт	168909000.0 нс	464608000.0 нс
20%	602004 байт	2000000 байт	602004 байт	2000000 байт	337620000.0 нс	441352000.0 нс

Процент заполнения	Объем памяти (разреженная_1)	Объем памяти (стандартная_1)	Объем памяти (разреженная_2)	Объем памяти (стандартная_2)	Время (разреженная)	Время (стандартная)
30%	902004 байт	2000000 байт	902004 байт	2000000 байт	511711000.0 нс	455917000.0 нс
40%	1202004 байт	2000000 байт	1202004 байт	2000000 байт	650621000.0 нс	434234000.0 нс
50%	1502004 байт	2000000 байт	1502004 байт	2000000 байт	786538000.0 нс	437069000.0 нс
60%	1802004 байт	2000000 байт	1802004 байт	2000000 байт	902970000.0 нс	436925000.0 нс
70%	2102004 байт	2000000 байт	2102004 байт	2000000 байт	862775000.0 нс	448589000.0 нс
80%	2402004 байт	2000000 байт	2402004 байт	2000000 байт	813611000.0 нс	441907000.0 нс
90%	2702004 байт	2000000 байт	2702004 байт	2000000 байт	700608000.0 нс	466937000.0 нс
100%	3002004 байт	2000000 байт	3002004 байт	2000000 байт	624379000.0 нс	444769000.0 нс

Матрица 1000x1000

Процент заполнения	Объем памяти (разреженная_1)	Объем памяти (стандартная_1)	Объем памяти (разреженная_2)	Объем памяти (стандартная_2)	Время (разреженная)	Время (стандартная)
10%	1204004 байт	8000000 байт	1204004 байт	8000000 байт	1384828000.0 нс	3567899000.0 нс
20%	2404004 байт	8000000 байт	2404004 байт	8000000 байт	2818081000.0 нс	3586504000.0 нс
30%	3604004 байт	8000000 байт	3604004 байт	8000000 байт	4107498000.0 нс	3944650000.0 нс
40%	4804004 байт	8000000 байт	4804004 байт	8000000 байт	5298194000.0 нс	3814375000.0 нс
50%	6004004 байт	8000000 байт	6004004 байт	8000000 байт	6322070000.0 нс	3667999000.0 нс
60%	7204004 байт	8000000 байт	7204004 байт	8000000 байт	7055800000.0 нс	3571700000.0 нс
70%	8404004 байт	8000000 байт	8404004 байт	8000000 байт	7023446000.0 нс	3584693000.0 нс
80%	9604004 байт	8000000 байт	9604004 байт	8000000 байт	6360985000.0 нс	3790463000.0 нс
90%	10804004 байт	8000000 байт	10804004 байт	8000000 байт	5640059000.0 нс	3606521000.0 нс
100%	12004004 байт	8000000 байт	12004004 байт	8000000 байт	5042356000.0 нс	3559826000.0 нс

Оценка эффективности в зависимости от процента заполнения

Найдем эффективность обработки разреженной матрицы в %, используя формулу:

Эффективность = $(100 - (100 * \text{Время выполнения разреженной матрицы} / \text{Время выполнения стандартной матрицы}))$

Количество ненулевых элементов, %	Эффективность для разреженной матрицы, %
10	67,19453251
20	30,05369404
30	-2,7040823
40	-42,35743029
50	-61,41219132
60	-62,35767965
70	-66,84908619
80	-80,66563384
90	-90,14024402
100	-92,28845706

Вывод

Относительная эффективность обработки разреженных матриц и обычных матриц зависит от процентного заполнения матрицы нулями. При плотных матрицах эффективнее использовать обычные матрицы, так как для хранения разреженных матриц придется использовать не только массив элементов но и массивы для хранения их столбцов/строк и индексы начала, увеличивая необходимое количество памяти в 2 раза. Кроме того, при обработке матриц с большим количеством ненулевых элементов время обработки разреженных матриц превышает время обработки обычной из-за сложности доступа к элементу в разреженной матрице. Однако при обработке матрицы с небольшим процентом заполнения разреженная матрица показывают большую эффективность, из-за отсутствия обработки ненулевых элементов. Поэтому выбор способа хранения матрицы и алгоритмов обработки зависит от конкретных задач и ожидаемого процента заполнения в этой задаче.

Ответ на контрольные вопросы

1. Что такое разреженная матрица, какие схемы хранения таких матриц Вы знаете?
Разреженная матрица - это матрица, в которой большинство элементов равно нулю. Такие матрицы встречаются, например, в задачах обработки изображений, графов и систем линейных уравнений. Существуют различные методы хранения элементов матрицы в памяти. Например, линейный связный список, т.е. последовательность ячеек, связанных в определенном порядке. Существует диагональная схема хранения симметричных матриц, а также - связные схемы разреженного хранения. Связная схема хранения матриц, предложенная Кнудом, 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). Наиболее широко используемая схема хранения разреженных матриц - это схема, предложенная Чангом и Густавсоном, называемая: "разреженный строчный формат". Эта схема предъявляет минимальные требования к памяти и очень удобна при выполнении операций сложения, умножения матриц, перестановок строк и столбцов, транспонирования, решения систем линейных уравнений, при хранении коэффициентов в разреженных матрицах и т.п.
2. Каким образом и сколько памяти выделяется под хранение разреженной и обычной матрицы? Для хранения разреженной матрицы выделяется значительно меньше памяти по сравнению с обычной матрицей, т.к. хранятся только ненулевые элементы. Количество памяти, которое выделяется для хранения разреженной матрицы, зависит

от выбранной схемы хранения и структуры самой матрицы. Например для способов хранения CSR и CSS потребуется $(\text{sizeof}(\text{type_1}) + \text{sizeof}(\text{type_2})) * \text{elem_cnt} + \text{sizeof}(\text{type_2}) * \text{cnt}$, где type_1 - тип элемента матрицы, type_2 - тип для хранения индексов строк/столбцов матрицы, elem_cnt - количество ненулевых элементов, cnt - количество строк/столбцов матрицы. Обычная матрица, в свою очередь, выделяет память под каждый элемент матрицы вне зависимости от его значения, на нее будет выделено $n * m * \text{sizeof}(\text{type})$, где type - тип элемента матрицы.

3. Каков принцип обработки разреженной матрицы? Обработка разреженной матрицы отличается от обработки обычной матрицы. Основной принцип обработки разреженной матрицы - исключить операции с нулевыми элементами, чтобы снизить вычислительные затраты и использование памяти. Конкретный способ обработки зависит от способа хранения матрицы (CSR, CSS)
4. В каком случае для матриц эффективнее применять стандартные алгоритмы обработки матриц? От чего это зависит? Стандартные алгоритмы обработки матриц эффективнее применять, когда матрица имеет почти все ненулевые элементы и размеры матрицы не очень велики. Это зависит от плотности разреженной матрицы - чем больше ненулевых элементов, тем менее эффективным будет использование схемы хранения разреженной матрицы. Если матрица плотная или имеет малое количество ненулевых элементов, использование стандартных алгоритмов обработки матриц может быть более эффективным.