

Лабораторная работа №3. "Разреженные матрицы"

Студент Ларин Владимир - ИУ7-34Б

Описание условия задачи

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**;
 - связный список **JA**, в элементе **Nk** которого находится номер компонент **A** и **IA**, с которых начинается описание столбца **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

Техническое задание

Входные данные:

1. Целое число, представляющее собой номер команды: целое число в диапазоне от 0 до 10.
2. Командно-зависимые данные:
 - количество строк/столбцов матрицы, количество ненулевых элементов матрицы, элементы матрицы в формате "индекс строки - индекс столбца – значение элемента", нумерация строк/столбцов начинается с единицы
 - количество строк/столбцов матрицы, процент ненулевых элементов матрицы.
 - количество столбцов вектора, количество ненулевых элементов вектора, элементы вектора в формате "индекс столбца – значение элемента", нумерация столбцов начинается с единицы
 - количество столбцов вектора, процент ненулевых элементов вектора.

Выходные данные:

1. Исходные матрица, исходный и результирующий вектора в стандартном виде и разреженном виде.
2. Количественная характеристика сравнения вариантов умножения вектора на матрицу.

Команды программы

- Работа с матрицей
1. Ввести матрицу
 2. Сгенерировать случайную матрицу
 3. Редактировать матрицу в VIM

4. Напечатать матрицу

- Работа с вектором

5. Ввести вектор

6. Сгенерировать случайный вектор

7. Редактировать вектор в VIM

8. Напечатать вектор

- Операция умножения

9. Умножить

10. Провести сравнение разных преставлений матриц

Обращение к программе: запускается из терминала.

Аварийные ситуации:

1. Некорректный ввод номера команды.

На входе: число, большее чем 10 или меньшее, чем 0.

На выходе: сообщение об ошибке

2. Некорректный ввод количества строк или столбцов матрицы.

На входе: неположительное целое число или буква.

На выходе: сообщение об ошибке

3. Некорректный ввод индекса строки или столбца матрицы/вектора.

На входе: число , выходящее за границы [1; количество_столбцов(строк) -1], или буква.

На выходе: сообщение об ошибке

4. Некорректный ввод элемента матрицы/вектора.

На входе: число, выходящее за границы разрядной сетки, или буква.

На выходе: сообщение об ошибке

Структуры данных

Для хранения элементов матриц используется тип `matrix_type_t`, который является псевданимом `double`

```
#define MATRIX_TYPE_SPECIF "lg"
typedef double matrix_type_t;
```

За стандартное хранение матрицы отвечает именованная структура `matrix_t`, описанная как:

```
typedef struct matrix {
    size_t rows; // Количество строк
    size_t cols; // Количество столбцов
    matrix_type_t **data; // Буффер данных
} matrix_t;
```

За хранение матрицы в разреженном столбцовом виде отвечает именованная структура `sparse_matrix_t`, описанная как:

```
typedef struct sparse_matrix {
    size_t rows; // Количество строк
    size_t cols; // Количество столбцов
    size_t count; // Количество ненулевых элементов
    matrix_type_t *elements; // Массив элементов
    size_t *elements_rows; // Массив строк элементов
    size_t *cols_starts; // Массив индексов элементов,
                        // с которых начинается столбец
} sparse_matrix_t;
```

Алгоритм

1. Пользователь вводит номер команды из меню.
2. Пока пользователь не введет 0 (выход из программы), ему будет предложено выполнять действия с матрицами.
 - Алгоритм случайной генерации матрицы/вектора $N \times N$ элементами
 1. Первые $N \times N$ элементов матрицы, при просмотре по строкам заполняются случайными элементами
 2. Матрица перемешивается
 1. При просмотре матрицы по строкам текущий элемент меняется со случайным элементом, следующим за текущим
 - Алгоритм перемножения стандартных матриц
 1. Создается нулевая выходная матрица нужного размера
 2. Просматривая все элементы выходной матрицы заполняю их по формуле $\sum_{k=1}^N \{A_{i,k} \cdot B_{k,j}\}$, где i и j - строка и столбец просматриваемого элемента
 - Алгоритм перемножения разреженных вектора и матрицы
 1. Создается пустой результирующий вектор
 2. Матрица просматривается по столбцам
 3. Каждый элемент i -й строки умножается на i -й элемент вектора, если он существует, и данная сумма добавляется к j -му элементу результирующего вектора, где j - текущий просматриваемый вектор

Описание основных функций

- `init_matrix(rows, cols);`
 - инициализирует матрицу

- принимает кол-во строк и столбцов
- возвращает созданную матрицу
- `free_matrix(matrix);`
 - освобождает матрицу
 - принимает матрицу
- `init_sparse_matrix(rows, cols, count);`
 - инициализирует разреженную матрицу
 - принимает кол-во строк, столбцов и ненулевых элементов матрицы
- `free_sparse_matrix(matrix);`
 - освобождает разреженную матрицу
 - принимает разреженную матрицу
- `mul(left, right, time);`
 - умножает две матрицы
 - принимает две матрицы
 - возвращает результат и время
- `mul_sparse(left, right, time);`
 - умножает разреженные вектор-строку на матрицу
 - принимает разреженные вектор строку и матрицу
 - возвращает результат и время
- `matrix_to_sparse(matrix)`
 - конвертирует стандартный формат матриц в разреженный
 - принимает стандартную матрицу
 - возвращает разреженную матрицу
- `sparse_to_matrix(sparse)`
 - конвертирует разреженный формат матриц в обычный
 - принимает разреженную матрицу
 - возвращает стандартную матрицу

Тесты

Описание теста	Ввод	Вывод
Неправильный номер команды	-1	Сообщение об ошибке
Неправильный номер команды	11	Сообщение об ошибке
Неправильный размер матрицы	1 0 0 0	Сообщение об ошибке
Неправильный номер элемента	1 1 1 1 0 0 1	Сообщение об ошибке
Неправильный номер элемента	1 1 1 1 2 2 1	Сообщение об ошибке
Неправильный размер вектора	5 0 0	Сообщение об ошибке
Неправильный номер элемента вектора	5 1 1 0 1	Сообщение об ошибке
Неправильный номер элемента вектора	5 1 1 2 1	Сообщение об ошибке

Описание теста	Ввод	Вывод
Неправильная заполненность матрицы	2 1 1 101	Сообщение об ошибке
Неправильная заполненность вектора	6 1 101	Сообщение об ошибке
Неправильные размеры для умножения	1 1 1 0 5 2 0 9	Сообщение об ошибке
Анализ	10	Сообщение об ошибке
Удачный ввод матрицы	1 10 10 1 2 2 5 4	Матрица размером 10x10 с одним элементом
Удачный ввод вектора	5 10 1 2 8	Вектор 10x10 с одним элементом
Удачное перемножение	1 2 2 2 1 1 4 2 2 5 5 2 1 2 55 9	Строка 1x2 с одним элементом 275

Оценка эффективности

Для каждого измерения взято среднее значение по времени для 100 разных случайных пар матрица-вектор.

- Время работы указано в тактах.
- Объём матриц указан в байтах.
- Заполненность указана в процентах

Размер матрицы	Заполненность матриц	Время умножения стандартных матриц	Время умножения разреженных матриц	Объём стандартных матриц	Объём разреженных матриц
16X16	0	3	2	2224	272
16X16	10	4	3	2224	688
16X16	20	4	4	2224	1136
16X16	30	3	4	2224	1552
16X16	40	4	4	2224	2000
16X16	50	4	5	2224	2448
16X16	60	3	6	2224	2864
16X16	70	3	6	2224	3312
16X16	80	3	6	2224	3728
16X16	90	3	6	2224	4176

Размер матрицы	Заполненность матриц	Время умножения стандартных матриц	Время умножения разреженных матриц	Объём стандартных матриц	Объём разреженных матриц
16X16	100	4	6	2224	4624
32X32	0	12	2	8496	528
32X32	10	11	4	8496	2208
32X32	20	9	4	8496	3888
32X32	30	9	5	8496	5584
32X32	40	11	7	8496	7264
32X32	50	11	10	8496	8976
32X32	60	11	13	8496	10656
32X32	70	10	12	8496	12336
32X32	80	10	13	8496	14032
32X32	90	8	13	8496	15712
32X32	100	7	9	8496	17424
64X64	0	28	2	33328	1040
64X64	10	26	5	33328	7680
64X64	20	26	9	33328	14336
64X64	30	41	18	33328	20992
64X64	40	35	23	33328	27648
64X64	50	29	30	33328	34320
64X64	60	37	39	33328	40960
64X64	70	38	44	33328	47616
64X64	80	28	36	33328	54272
64X64	90	29	35	33328	60928
64X64	100	36	48	33328	67600
128X128	0	120	2	132144	2064
128X128	10	115	15	132144	28464
128X128	20	113	31	132144	54880
128X128	30	116	54	132144	81312
128X128	40	132	90	132144	107728

Размер матрицы	Заполненность матриц	Время умножения стандартных матриц	Время умножения разреженных матриц	Объём стандартных матриц	Объём разреженных матриц
128X128	50	129	128	132144	134160
128X128	60	102	120	132144	160560
128X128	70	114	140	132144	186976
128X128	80	115	148	132144	213408
128X128	90	122	156	132144	239824
128X128	100	127	163	132144	266256
256X256	0	451	3	526384	4112
256X256	10	463	56	526384	109360
256X256	20	498	126	526384	214640
256X256	30	464	213	526384	319888
256X256	40	441	321	526384	425168
256X256	50	478	445	526384	530448
256X256	60	505	553	526384	635696
256X256	70	461	556	526384	740976
256X256	80	485	575	526384	846224
256X256	90	447	558	526384	951504
256X256	100	442	560	526384	1056784
512X512	0	1936	6	2101296	8208
512X512	10	1757	181	2101296	428448
512X512	20	1704	445	2101296	848688
512X512	30	1691	780	2101296	1268944
512X512	40	1764	1282	2101296	1689184
512X512	50	1696	1695	2101296	2109456
512X512	60	1945	2117	2101296	2529696
512X512	70	1859	2180	2101296	2949936
512X512	80	1707	2097	2101296	3370192
512X512	90	1748	2090	2101296	3790432
512X512	100	1708	2084	2101296	4210704

Размер матрицы	Заполненность матриц	Время умножения стандартных матриц	Время умножения разреженных матриц	Объём стандартных матриц	Объём разреженных матриц
1024X1024	0	6886	10	8396848	16400
1024X1024	10	6615	692	8396848	1695744
1024X1024	20	6603	1771	8396848	3375104
1024X1024	30	6709	3088	8396848	5054464
1024X1024	40	6645	4828	8396848	6733824
1024X1024	50	7134	6893	8396848	8413200
1024X1024	60	6723	7741	8396848	10092544
1024X1024	70	6616	8161	8396848	11771904
1024X1024	80	6857	8485	8396848	13451264
1024X1024	90	6963	8554	8396848	15130624
1024X1024	100	6826	8257	8396848	16810000
2048X2048	0	26611	19	33570864	32784
2048X2048	10	26518	2754	33570864	6746928
2048X2048	20	26453	6819	33570864	13461088
2048X2048	30	26930	12430	33570864	20175264
2048X2048	40	26740	19756	33570864	26889424
2048X2048	50	27182	27445	33570864	33603600
2048X2048	60	26808	31334	33570864	40317744
2048X2048	70	26943	33409	33570864	47031904
2048X2048	80	27090	34063	33570864	53746080
2048X2048	90	27330	33842	33570864	60460240
2048X2048	100	26953	33000	33570864	67174416

Контрольные вопросы

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

Разреженная матрица – это матрица, содержащая большое количество нулей. Способы хранения: связная схема хранения, строчный формат, столбцовый формат, линейный связный список, кольцевой связный список, двунаправленные стеки и очереди.

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 $2 * K + M$ ячеек памяти, где K – количество ненулевых элементов.

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 алгоритмы обработки матриц эффективнее применять при большом количестве ненулевых элементов (от 50%).

Вывод

Например, для заполненности 20% матрицы 2048x2048 разреженный формат выполняется на 288% быстрее и при этом объём памяти на 149% меньше. А для процента 50% заполненности разреженный формат сравним с обычным форматом по времени работы и по объёму занимаемой памяти. Для процента заполненности 100% обычный метод быстрее на 22%, и объём занимаемой памяти меньше в 2 раза.

Значит, при заполненности матрицы менее 50% для задачи умножения вектора строки на матрицу целесообразнее использовать метод хранения разреженных матриц, иначе следует использовать метод стандартного хранения.