

Условие 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;
- вектор JA, в элементе Nk которого находится номер компонент в A и IA, с которых начинается описание столбца Nk матрицы A.

1. Смоделировать операцию умножения вектора-строки хранящегося в форме вектора A и вектора, содержащего номера столбцов этих элементов, и матрицы, хранящейся в указанной форме, с получением результата в форме хранения вектора-строки.

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:

1. Программа обрабатывает целые числа, поэтому все входные данные – целые числа.

2. Размерности матриц больше 0.

3. Заполнение массивов осуществляется пока пользователь не введет 0 в массив ненулевых элементов.

4. Ввод осуществляется по столбцам: элемент 0-го столбца, его строка и столбец, следующий элемент этого столбца и так далее.

5. В файле данные хранятся так: 1-я строка – размерности матрицы, вторая строка – ненулевые элементы матрицы, 3-я строка их строки, 4-я их столбцы, 5-я ненулевые элементы вектора-строки, 6-я – их столбцы.

6. Строки в файле заканчиваются символом '\n'.

7. Нумерация столбцов/строк начинается от нуля до количества столбцов/строк – 1.

Алгоритм:

1. Программа выводит краткую информацию по тому, как она работает.

2. Пользователю предлагается выбрать метод ввода матрицы, вручную или из файла.

3. Программа получает данные, заполняет координатное представление матрицы и вектора строки.

4. Создается представление введенных матриц в “нормальном” виде.

5. Перемножаются вектор строка и матрица в обычном виде.

6. Создается разреженное представление матрицы.
7. Перемножается вектор-строка в представлении: ненулевые элементы и их столбцы, на матрицу в разреженном представлении.
8. Выводятся результаты умножения каждым способом, замеры времени, памяти.

Типы данных:

В программе используются 3 структуры:

```
MAX_LEN = 10000
```

```
typedef struct
{
    int a[MAX_LEN + 1];
    int ia[MAX_LEN + 1];
    int ja[MAX_LEN + 1];
    int len_a;
    int rows;
    int columns;
} coord;
```

```
typedef struct
{
    int matrix[MAX_LEN / 100][MAX_LEN / 100];
    int rows;
    int columns;
} reg_matrix;
```

```
typedef struct
{
    int a[MAX_LEN + 1];
    int ia[MAX_LEN + 1];
    int ja[MAX_LEN + 1];
    int len_a;
    int rows;
    int columns;
} sparse;
```

Первая структура описывает координатную форму записи матрицы, вторая – стандартную форму записи, третья – разреженную.

Тесты:

Входные данные	Выход программы	Код возврата
Нет файла	Ошибка! Невозможно открыть файл.	15
Файл пустой	Ошибка! Файл пустой.	17
Неверные размерности матрицы	Ошибка! Неправильно введены размерности матрицы.	4
Некорректные данные в ненулевом массиве	Ошибка! Неправильно введен ненулевой элемент.	1
Некорректные данные в массиве строк	Ошибка! Неправильно введена строка элемента.	2
Некорректные данные в массиве столбцов	Ошибка! Неправильно введен столбец элемента.	3
Некорректные данные в ненулевых элементах вектора	Ошибка! Неправильно введен ненулевой элемент.	1
Некорректные данные в массиве столбцов вектора	Ошибка! Неправильно введен столбец элемента.	3
Неверные размерности в файле	Ошибка! Неправильные размерности матрицы в файле.	5
Некорректные данные в ненулевом массиве в файле	Ошибка! Неверный формат массива ненулевых элементов в файле.	6
Некорректные данные в массиве строк в файле	Ошибка! Неверный формат массива строк элементов в файле.	8

Некорректные данные в массиве столбцов в файле	Ошибка! Неверный формат массива столбцов элементов в файле.	9
Некорректные данные в ненулевых элементах вектора в файле	Ошибка! Неверный формат массива ненулевых элементов в файле.	6
Некорректные данные в массиве столбцов вектора в файле	Ошибка! Неверный формат массива столбцов элементов в файле.	9
Неверно введено имя файла	Ошибка! Неправильно введено имя файла.	10
Матрица пустая	Ошибка! Матрица пустая.	7
Неверная длина массива ненулевых элементов в файле	Ошибка! Неверная длина массива ненулевых элементов в файле.	11
Неверная длина массива строк в файле	Ошибка! Неверная длина массива строк элементов в файле	12
Неверная длина массива столбцов в файле	Ошибка! Неверная длина массива столбцов элементов в файле	13
Неверная длина массива ненулевых элементов вектора-строки в файле	Ошибка! Неверная длина массива ненулевых элементов вектора-строки в файле.	14
Неверная длина массива столбцов элементов вектора-строки в файле	Ошибка! Неверная длина массива столбцов элементов вектора-строки в файле.	18
Матрица 20 на 20 заполнена на 1 процент	Результат умножения в координатной форме , замеры времени и памяти	0

Матрица 20 на 20 заполнена на 10 процентов	Результат умножения в координатной форме , замеры времени и памяти	0
Матрица 20 на 20 заполнена на 30 процентов	Результат умножения в координатной форме , замеры времени и памяти	0
Матрица 20 на 20 заполнена на 50 процентов	Результат умножения в координатной форме , замеры времени и памяти	0
Матрица 20 на 20 заполнена на 65 процентов	Результат умножения в координатной форме , замеры времени и памяти	0
Матрица 20 на 20 заполнена на 80 процентов	Результат умножения в координатной форме , замеры времени и памяти	0
Матрица 20 на 20 заполнена на 100 процентов	Результат умножения в координатной форме , замеры времени и памяти	0

Результаты замеров времени и памяти:

Замеряем умножение матрицы размерностью 100x100 на вектор длиной 100, вектор заполнен на 100%.

Процент заполнения матрицы	Время умножения стандартным способом в секундах	Время умножения разреженных матриц в секундах	Память затраченная на стандартное умножение	Память затраченная на умножение разреженных матриц
1	0.008165	0.000274	40824	3184
10	0.008074	0.002583	40824	10828
30	0.008334	0.004356	40824	26828
50	0.007997	0.008759	40824	42828
70	0.008075	0.010957	40824	58828
90	0.008126	0.016478	40824	74828
100	0.012086	0.030465	40824	82828

Вывод:

Из результатов понятно что умножение разреженных матриц эффективнее, чем умножение стандартным способом, вплоть до заполнения матрицы на 50%, затем на такое умножение тратится слишком много памяти и времени по сравнению со стандартным перемножением, поэтому, в случаях заполнения матриц больше чем на 50% лучше использовать стандартное умножение, а если матрица все-таки разреженная, легче использовать метод перемножения разреженных матриц.

Контрольные вопросы:

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

Разреженная матрица – это матрица в которой нулевых элементов больше чем 50%.

Схемы хранения матриц:

- Связная схема хранения матриц, предложенная Кнутом
- Схема, предложенная Чангом и Густафссоном, называемая "разреженный строчный формат"

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

Для обычной матрицы память выделяется для хранения всех элементов, в том числе и нулевых. Количество памяти, выделяемой под разреженную матрицу зависит от количества ненулевых элементов.

3. Каков принцип обработки разреженной матрицы?

В разреженной матрице мы обрабатываем только ненулевые элементы.

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

Разреженная матрица эффективна, только когда в массиве много нулевых элементов.

Если их незначительное количество, то намного удобнее использовать стандартный способ представления и обработки матриц.