­­­­Лабораторная работа 5

Обработка разреженных матриц

Монахов Дмитрий ИУ7-34

Отчет (Вариант 13, задание 6)

**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***;

- связный список ***JA***, в элементе Nk которого находится номер компонент

в ***A*** и ***IA***, с которых начинается описание столбца Nk матрицы ***A***.

1. Смоделировать операцию сложения двух матриц, хранящихся в этой форме, с получением результата в той же форме.

2. Произвести операцию сложения, 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**

**Аварийные ситуации**

Ошибка выделения памяти

Некорректная команда пользователя

**Предусмотрено**

Сообщение об ошибке выделения памяти

Сообщение о некорректной команде

**Входные данные**

Команда пользователя

Данные для заполнения матриц

**Выходные данные**

Содержимое матриц

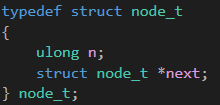
Результат сложения матриц

Результаты анализа

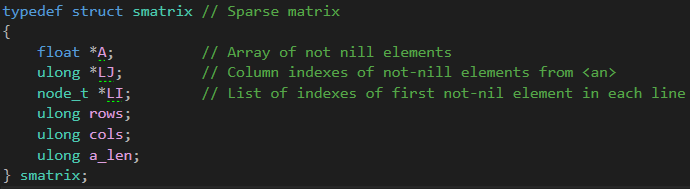
Сообщение об ошибке (если имеется)

**Структуры данных**

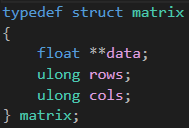
1. Элемент списка



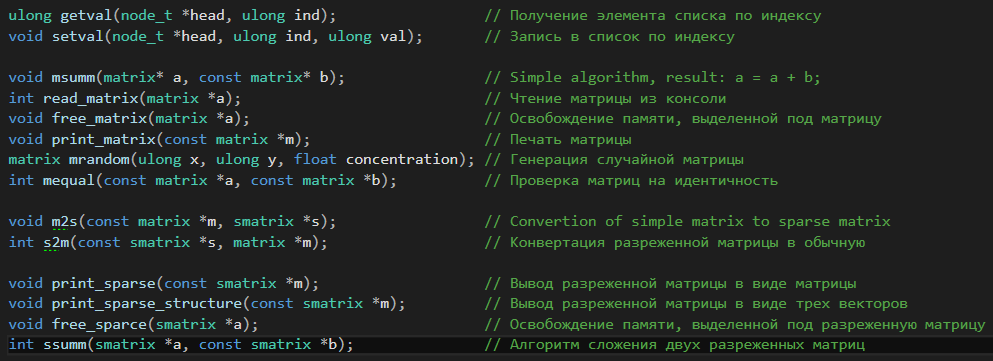
1. Разреженная матрица



1. Обычная матрица



**Функции**



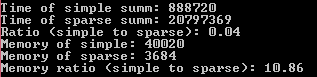
**Тесты**

1. Некорректная команда пользователя
2. Заполненность матрицы 0%

**Анализ**

Эффективность применения разреженной матрицы тем больше, чем больше в матрице нулевых элементов. Для этого запустим программу на матрицах с разными размерами. Будем последовательно увеличивать количество нулевых элементов, увеличивая размер и уменьшая показатель заполненности.

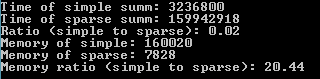
Сложение матриц 100 x 100 с заполненностью 1%:



Видно, что по времени сложение обычных матриц эффективнее в 23 раза.

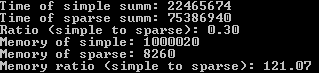
Однако даже на этом этапе разреженная матрица требует в 10,86 раза меньше памяти чем обычная. Продолжим уменьшать концентрацию элементов.

Сложение матриц 200 x 200 с заполненностью 0,5%:



Разница во времени выросла до 50 раз, по памяти до 20,44 раза.

Сложение матриц 500 x 500 с заполненностью 0,001%

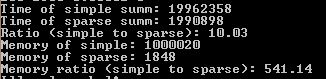


Разреженная матрица эффективнее по памяти уже в 121 раз, однако по времени она все еще проигрывает. Однако возможно подобрать такие исходные данные, при которых разреженная матрица будет эффективнее и по времени.

В связи с тем, что третий вектор представлен связным списком, большое количество времени затрачивается на его просмотр. Ситуация усугубляется тем что этот вектор всегда содержит столько элементов, сколько в матрице строк вне зависимости от того, есть ли в соответствующей строке элементы.

Получается, что можно создать матрицу с малым количеством строк, но таким же количеством элементов. Попробуем повторить эксперимент с 500 \* 500 элементами, но с другим размером матрицы.

Сложение матриц 100 x 2500 с заполненностью 0,001%



Видно, что в этом случае сложение разреженных матриц эффективнее обычных в 10 раз по времени и в 541 раз по памяти.

**Вывод**

Разреженные матрицы эффективнее обычных по памяти, когда нулевых элементов достаточно много. Однако чтобы добиться эффективности по времени нужна очень большая степень разреженности. Эксперимент показал, что на выбранных мною алгоритмах разреженные матрицы становятся эффективнее обычных по времени, когда из 40000 элементов ненулевых всего 4.