**Условие задачи:** реализовать алгоритмы обработки разреженных матриц, сравнить эффективность использования этих алгоритмов (по времени выполнения и по требуемой памяти) со стандартными алгоритмами обработки матриц при различном процентном заполнении матриц ненулевыми значениями.

**Описание ТЗ**: реализовать работу с разреженными матрицами. Разреженная 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.

Выполнить следующие задания:

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.

**Возможные ошибки:**

- Неверный выбор операции

- Неверный ввод целочисленных переменных

- Попытка ввода за границы массива

**Описание алгоритма:**

Цикл от 1-го столбца до последнего

начало

Суммирование элементов 1-й матрицы к результирующему массиву;

Суммирование элементов 2-й матрицы к результирующему массиву;

конец;

**Описание структуры данных:**

Разреженная матрица:

int \*A; // Массив ненулевых элементов

int \*IA; // Массив строк

struct SpType \*last; // Список столбцов

struct SpType

{

int value;

struct SpType\* NextAdr;

};

**Сравнение производительности:**

Разреженность 60%

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 100 | 200 | 300 | 400 | 500 | 900 |
| Усовершенствованное сложение | 1009581 | 7450319 | 23577846 | 54576292 | 106069824 | 603120024 |
| Обычное сложение | 68586 | 263316 | 564888 | 985702 | 1511940 | 5556309 |

Разреженность 80%

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 100 | 200 | 300 | 400 | 500 | 900 |
| Усовершенствованное сложение | 304137 | 2039856 | 6443181 | 14736570 | 27795186 | 155447138 |
| Обычное сложение | 66885 | 255300 | 536133 | 903996 | 1442664 | 5410371 |

Разреженность 97%

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 100 | 200 | 300 | 400 | 500 | 900 |
| Усовершенствованное сложение | 17754 | 87891 | 231597 | 487689 | 909300 | 4448394 |
| Обычное сложение | 65289 | 237699 | 516270 | 916575 | 1451664 | 5418744 |

Малое количество элементов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 10 | 20 | 30 | 40 |
| 60% | 2763 | 12822 | 34386 | 68568 |
| 80% | 2412 | 7509 | 11145 | 27906 |
| 89% | 1370 | 2988 | 6261 | 11280 |
| Обычное сложение | 1874 | 3084 | 6639 | 12072 |

По этим данным видно, что обычный алгоритм сложения эффективен при малой и средней разреженности матриц. При разреженности 97% обычный алгоритм становится менее эффективным. Связано это с тем, что при усовершенствованном сложении нам приходится изменять сразу 3 объекта. Что касается требуемой памяти, обычный алгоритм сложения становится не эффективным при большой разреженности, так как приходится обрабатывать большое количество нулевых элементов. При малой разреженности он более эффективен, так как траты на хранение 3-х объектов в таком случае довольно большие. При малом количестве элементов усовершенствованный метод эффективнее при разреженности 89%.

**Контрольные вопросы:**

1. Разреженная матрица – матрица, содержащая много нулей. Хранить матрицы можно в виде линейного связного списка. Также можно хранить матрицу, используя кольцевой связный список, двунаправленные стеки и очереди. Существует диагональная схема хранения симметричных матриц, а также связные схемы разреженного хранения.
2. При хранении обычной матрицы память выделяется под все элементы, в том числе и нулевые. При хранении разреженной матрицы память выделяется только под ненулевые элементы.
3. Принцип обработки заключается в том, что мы просматриваем только ненулевые элементы и осуществляем работу с ними.
4. Обычный алгоритм сложения эффективен при малой и средней разреженности матриц. При разреженности 97% обычный алгоритм становится менее эффективным. Связано это с тем, что при усовершенствованном сложении нам приходится изменять сразу 3 объекта. Что касается требуемой памяти, обычный алгоритм сложения становится не эффективным при большой разреженности, так как приходится обрабатывать большое количество нулевых элементов. При малой разреженности он более эффективен, так как траты на хранение 3-х объектов в таком случае довольно большие. При малом количестве элементов усовершенствованный метод эффективнее при разреженности 89%.