Типы и Структуры Данных

Лабораторная работа №3

*Работа с разряженной матрицей*

Выполнил студент группы: ИУ7-36

Хеламбаге Гавинду Саджиндра Де Силва

# Задание:

Разреженная (содержащая много нулей) матрица хранится в форме 3-х объектов:

- вектор ***A*** содержит значения ненулевых элементов;

- вектор ***JA*** содержит номера столбцов для элементов вектора ***A***;

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

в ***A*** и ***JA***, с которых начинается описание строки Nk матрицы ***A***.

Смоделировать операцию умножения матрицы и вектора-столбца, хранящихся в этой форме, с получением результата в той же форме.

# ТЗ:

Целью работы:- является реализация алгоритмов обработки разреженных матриц.

# Входные данные.

Первоначально пользователь получает запуск программы с введенными вручную матрицами. В случае запуска программы пользователю предлагается ввести размер матриц: размер матрицы вводится один за другим для строки и для столбца соответственно.

# Выходные данные.

# В процессе выполнения программа отображает созданные пользователем матрицы в случае ручного ввода. В результате работы программа отображает полученную матрицу - сумму из 2 указанных матриц.

# Обработка ошибок.

Если команда запуска введена неправильно, программа сообщает пользователю об ошибке и предлагает ввести команду еще раз.

# Алгоритм работы и используемые структуры:

**Алгоритм:**

Вы вводили в матрицы вручную, пользователю предлагается ввести количество ненулевых элементов (сначала для 1 матрицы, затем для 2), а также сами элементы (ввод выполняется в 3 этапа: значение элемента, координата на линии , почтой).

**Структуры данных:**

В программе была использована структура матриц, хранящая в себе массив элементов, массив индексов по I, и также список, в котором хранится индекс первого элемента по строке.

struct ElementNode {

int index;

struct ElementNode \*next;

};

struct SparseMatrix { int \*A;

int \*IA;

struct ElementNode \*JA; int size;

int rows; int columns;

};

# 4.Тесты:

# 

# 

# 5.Контрольные вопросы:

1. Что такое разреженная матрица, какие схемы хранения таких матриц Вы знаете?
   * Разреженная матрица - матрица, в которой есть определенный процент не нулевых элементов, а все остальные элементы = 0. Наиболее широко используемая схема хранения разреженных матриц - это схема, предложенная Чангом и Густавсоном, называемая: "разреженный строчный формат". В этом случае значения ненулевых элементов хранятся в массиве AN, соответствующие им столбцовые индексы - в массиве JA. Кроме того, используется массив указателей, например IA, отмечающих позиции AN и JA, с которых начинаются описание очередной строки. Дополнительная компонента в IA содержит указатель первой свободной позиции в JA и AN.
2. Каким образом и сколько памяти выделяется под хранение разреженной и обычной матрицы?
   * При хранении обычной матрицы память выделяется под каждый элемент (включая нулевые). В случае с разреженной матрицей, память выделяется только под не нулевые элементы. Значения ненулевых элементов хранятся в массиве AN, соответствующие им столбцовые индексы - в массиве JA. Кроме того, используется массив указателей, например IA, отмечающих позиции AN и JA, с которых начинаются описание очередной строки. Дополнительная компонента в IA содержит указатель первой свободной позиции в JA и A. Поэтому при процентном заполнении матрицы = 100%, хранение матрицы в разреженном виде займет больше места, чем хранение матрицы в обычном виде.
3. Каков принцип обработки разреженной матрицы?
   * Нет необходимости пробегать по всей матрице, проверяя каждый ее элемент. Нам заранее известны ненулевые элементы и их позиции, поэтому работая с разреженной матрицей мы работаем только с ненулевыми элементами (это значительно ускоряет работу программы).
4. В каком случае для матриц эффективнее применять стандартные алгоритмы обработки матриц? От чего это зависит?
   * Стандартные алгоритмы обработки эффективно применять в случае большого процента заполнения или маленького размера матриц. В случае, если заданы большие матрицы с низким процентом ненулевых элементов, выгодно использовать разреженные матрицы и алгоритмы работы с ними.