Московский Авиационный Институт
(Национальный Исследовательский Университет)

Факультет прикладной математики и информатики
Кафедра вычислительной математики и программирования

Курсовой проект
По дисциплине
«Практикум на ЭВМ»
2 семестр

Задание VII
«Разреженные матрицы»

Студент:	Сикорский А. А.
Группа:	М8О-108Б-20
Преподаватель:	Трубченко Н. М.
Подпись:	
Оценка:	
Дата:	20.05.2021

Москва
2021

Содержание

Задание

Составить программу на языке Си с функциями для обработки разреженных матриц с элементами целого типа, которая:

1. вводит матрицы различного размера, представленные во входном текстовом файле в обычном формате (по строкам), с одновременным размещением ненулевых элементов в разреженной матрице в соответствии с заданной схемой;

2. печатает введенные матрицы во внутреннем представлении согласно заданной схеме размещения и в обычном (естественном) виде;

3. выполняет необходимые преобразования разреженных матриц (или вычисления над ними) путем обращения к соответствующим функциям;

4. печатает результат преобразования (вычисления) согласно заданной схеме размещения и в обычном виде.

1 Программа

1.1 Структура данных

Матрица работает на трех векторах, соответственно в программе присутствует реализация вектора. У вектора есть следующие функции:

1. **resize** изменяет размер вектора, увеличивая текущий в два раза. Работает через *realloc*, ведь нам нужно довыделить или найти и скопировать значения в **последовательный** участок памяти.

2. **size** возвращает размер вектора, который поступил на вход фукнции в качестве аргумента.

3. **push** записывает элемент в вектор по заданному индексу. Если полученнный индекс больше текущего размера вектора, вызывается *resize*. Существует некоторый запас для добавления новых элементов, чтобы не вызывать *resize*, а соответственно не обращаться к системе за памятью на каждое добавление элемента. Помогает в быстродействии структуры.

4. **create** инициализация пустого вектора размера *size*. В ячейках памяти по индексам лежит не мусор, а нули.

5. **read** читает элемент вектора по индексу, проверяет выход за границу вектора. В STL C++ можно читать элемент снаружи непосредственно по индексу, избегая проверки

принадлежности этого индекса вектору. Это немного экономит время работы за счет отсуствия проверок, но позволяет натыкаться на ошибки.

6. **destroy** очищает и удаляет вектор.

7. **print** печатает вектор целиком.

1.2 Интерфейс и и выполнение задания

Курсовая работа не предполагает интерфейса. В задании не нужно вручную изменять элемент матрицы. Матрица считывается из файла, который поступает аргументом из командной строки. Если файл открыть не получилось, программа закрывается через *exit()*. Если всё хорошо и доступ к файлу получен, из него считывается размер матрицы и далее сама матрица.

Листинг 1: Открытие и считывание файла

```
int main(int argc, char* argv[]) {
FILE* f;
.
.
.
if (!(f = fopen(argv[1], "r"))) {
    printf("wrong file");
    exit(1);
}
else
.
.
.
```

Задание: Транспонировать разреженную матрицу относительно побочной диагонали. Выяснить, является ли полученная матрица кососимметрической.

По определению кососимметрической называют **квадратную** матрицу A такую, что $A^T = -A$. Например матрица

$$A = \begin{pmatrix} 0 & -2 & -1 \\ 2 & 0 & 3 \\ 1 & -3 & 0 \end{pmatrix}$$

является кососимметрической.

После транспонирования матрицы исходная матрица удаляется, а на кососимметричность проверяется её транспонированная копия. При проверке сразу отбрасываются любые матрицы, отличные от квадратных. Квадратная нулевая матрица по определению же считается кососимметрической. Некоторые тесты:

$$A = \begin{pmatrix} 0 & -2 & -1 \\ 2 & 0 & 3 \\ 1 & -3 & 0 \end{pmatrix}$$

3 3

0 -2 -1

2 0 3

1 -3 0

matrix before::

v 1: 0 2 4

v 2: 2 3 1 3 1 2

v 3: -2 -1 2 3 1 -3

0 -2 -1

2 0 3

1 -3 0

matrix after::

v 1: 0 2 4

v 2: 2 3 1 3 1 2

v 3: 3 -1 -3 -2 1 2

0 3 -1

-3 0 -2

1 2 0

symmetric

$$A = \begin{pmatrix} 1 & 2 & 3 \end{pmatrix}$$

3 1

1 2 3

matrix before::

v 1: 0 1 2

v 2: 1 1 1

v 3: 1 2 3

1

2

3

matrix after::

v 1: 0

v 2: 1 2 3

v 3: 3 2 1

3 2 1

not symmetric

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

matrix before::

v 1: -1 -1 -1

v 2:

v 3:

0 0 0

0 0 0

0 0 0

matrix after::

v 1: -1 -1 0

v 2:

v 3:

0 0 0

0 0 0

0 0 0

symmetric

Векторы, отвечающие за сами элементы в случае нулевых матриц пустые, ненулевые элементы отсутствуют.

2 Вывод

В ходе работы я составил программу на языке Си с функциями для обработки прямоугольных разреженных матриц с элементами целого типа, которая:

1. вводит матрицы различного размера, представленные во входном текстовом файле в обычном формате (по строкам), с одновременным размещением ненулевых элементов в разреженной матрице в соответствии с заданной схемой;

2. печатает введенные матрицы во внутреннем представлении согласно заданной схеме размещения и в обычном (естественном) виде;

3. выполняет необходимые преобразования разреженных матриц (или вычисления над ними) путем обращения к соответствующим функциям;

4. печатает результат преобразования (вычисления) согласно заданной схеме размещения и в обычном виде.

Хранение матрицы было реализовано на трех векторах так, что в первом хранятся индексы начала строк в двух других векторах, во втором хранятся номера столбцов, а в третьем хранятся сами значения матрицы по определенным ранее позициям. Нулевые элементы в векторах не хранятся, ведь в оптимизации хранения разреженных матриц и есть смысл работы.