



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени
Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ»

КАФЕДРА «ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭВМ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3 ПО ДИСЦИПЛИНЕ: ТИПЫ И СТРУКТУРЫ ДАННЫХ

Обработка разреженных матриц

Студент **Ширяев А.А.**

Группа **ИУ7-33Б**

Название предприятия **НУК ИУ МГТУ им. Н. Э. Баумана**

Студент _____ **Ширяев А.А.**

Преподаватель _____ **Силантьева. А.В**

2024

Лабораторная работа №3 по дисциплине “Типы и структуры данных”	1
Условие задачи	3
Описание техзадачи	4
a. Описание исходных данных	4
b. Описание задачи, реализуемой программой	5
c. Способ обращения к программе	5
d. Описание возможных аварийных ситуаций и ошибок пользователя	5
Описание внутренних СД	6
Информация о матрице (разреженный формат):	6
Информация о матрице (обычный формат):	7
Информация о векторе (формат вектор-строки):	7
Информация о векторе (обычный формат):	8
Описание алгоритма	8
Набор тестов	11
Таблицы с результатами измерений времени и памяти при различных используемых форматах хранения и алгоритмах обработки матриц в их различном процентном заполнении нулями.	14
Выводы по проделанной работе	27
Контрольные вопросы	27

Цель работы - 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 и при различных размерах матриц.

Условие задачи

Вариант 4

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.

Вектор-строка хранится в 2х объектах:

- вектор B, содержащий значения ненулевых элементов
- вектор JB, параллельный вектору B, содержащий индексы ненулевых элементов

1. Смоделировать операцию умножения вектора-строки и матрицы, хранящихся в указанной форме, с получением результата в форме хранения вектора.
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.

ПРИМЕЧАНИЕ! В отчёте будут использоваться обозначения для действий:

<init> - Инициализация матрицы и вектора

 <autoinit> - Автоматическая инициализация

 <self_init> - Заполнение вручную

 <self_autoinit> - Заполнение вручную только ненулевых элементов

 <init_coords> - Заполнение по координатам

<output> - Вывод матрицы и вектора

 <output_str_sparse> - Вывод матрицы в разреженном формате и вектор-строку

 <output_usual> - Вывод матрицы и вектора в обычном формате

<count vector_str> - Посчитать произведение, используя вектор-строку и матрицу в разреженном формате

<count vector> - Посчитать произведение, используя вектор и матрицу

<statistics> - Проведение замеров и вывод статистики об эффективности (по памяти, по времени выполнения) умножений.

<exit> - Выход из программы

(ПРИМЕЧАНИЕ! Если кол-во строк матрицы будет равно 1-му, матрица в программе будет представляться подобно вектору-строке. По этой причине в данном случае:

Матрица в обычном формате -> вектор в обычном формате

Матрица в разреженном формате -> вектор-строка

ЕСЛИ КОЛ-ВО СТРОК МАТРИЦЫ РАВНО 1)

Описание техзадачи

а. Описание исходных данных

Данные на входе: Меню. Код действия. Далее для каждого действия:

<init> -

<autoinit> - Размеры матрицы (больше нуля) (кол-во строк и столбцов), процент заполнения матрицы (1%-100%), процент заполнения вектора(1%-100%).

<self_init> - Размеры матрицы (больше нуля) (кол-во строк и столбцов), элементы матрицы, элементы вектора

<self_autoinit> - Размеры матрицы (больше нуля) (кол-во строк и столбцов), процент заполнения матрицы(1%-100%), процент заполнения вектора(1%-100%), элементы матрицы, элементы вектора

<init_coords> Размеры матрицы (больше нуля) (кол-во строк и столбцов), координаты элементов матрицы, координаты элементов вектора, элементы матрицы, элементы вектора

<output> -

<output_str_sparse> - Матрица в разреженном виде, вектор-строка

<output_usual> - Матрица в разреженном виде, вектор-строка

<count vector_str> - Матрица в разреженном виде, вектор-строка

<count vector> - Матрица в обычном формате, вектор в обычном формате

<statistics> - Размеры матрицы (больше нуля)

Данные на выходе:

<init> -

<autoinit> - Матрица в разреженном виде, вектор-строка, Матрица в обычном формате, вектор в обычном формате

<self_init> - Матрица в разреженном виде, вектор-строка, Матрица в обычном формате, вектор в обычном формате

<self_autoinit> - Матрица в разреженном виде, вектор-строка, Матрица в обычном формате, вектор в обычном формате

<output> -

<init_coords> Матрица в разреженном виде, вектор-строка, Матрица в обычном формате, вектор в обычном формате

<output_str_sparse> - Информация о матрице в разреженном виде, Информация о вектор-строке

<output_usual> - Информация о матрице в обычном формате, Информация о векторе в обычном формате

<count vector_str> - Информация о результате (векторе) в формате вектор-строка

<count vector> - Информация о результате (векторе) в обычном формате

<statistics> - Информация в форме таблицы с результатами замеров

b. Описание задачи, реализуемой программой

Программа реализует ряд действий:

<init> - Инициализация матрицы и вектора
 <autoinit> - Автоматическая инициализация
 <self_init> - Заполнение вручную
 <self_autoinit> - Заполнение вручную только ненулевых элементов
 <init_coords> - Заполнение по координатам
<output> - Вывод матрицы и вектора
 <output_str_sparse> - Вывод матрицы в разреженном формате и вектор-строку
 <output_usual> - Вывод матрицы и вектора в обычном формате
<count vector_str> - Посчитать произведение, используя вектор-строку и матрицу в разреженном формате
<count vector> - Посчитать произведение, используя вектор и матрицу
<statistics> - Проведение замеров и вывод статистики об эффективности (по памяти, по времени выполнения) умножений.
<exit> - Выход из программы

c. Способ обращения к программе

Для обращения к программе запускается файл *app.exe*.

d. Описание возможных аварийных ситуаций и ошибок пользователя

Программа может не вывести результат, а вывести сообщение об ошибке. Данная ситуация может произойти при условии:

Ошибки пользователя

1. Введен неверный код действия
2. Код действия содержит иные символы
3. <init> Введены некорректные размеры
4. <init>: <autoinit>, <self_autoinit> Введён некорректный процент
5. <init>: <self_init>, <self_autoinit>, <init_coords> Введён некорректный элемент
6. <init>: <self_init>, <self_autoinit> Введён нулевой элемент
7. <init>: <init_coords> Введена неправильная координата
8. <output> Информация о матрице и векторе отсутствует

- 9. <count vector_str> Информация о матрице и векторе отсутствует
- 10. <count vector> Информация о матрице и векторе отсутствует
- 11. <statistics> Введены некорректные размеры

Аварийные ситуации

- 1. Программа не смогла выделить необходимую память для работы

Описание внутренних СД

Информация о матрице (разреженный формат):

```
typedef struct
{
    int *A;
    size_t *IA;
    size_t *JA;

    size_t size;
    size_t JA_size;
} sparse_t;
```

Информация о матрице (разреженный формат) представляет собой структуру на языке Си, состоящую из:

- *A** - Указатель на память, содержащую значения ненулевых элементов
- *IA** - Указатель на память, содержащую номера строк для элементов вектора A
- *JA** - Указатель на память, содержащую номер компонента в A и IA, с которых начинается описание столбца Nk матрицы A.

size - Размер *A и *IA

JA_size - Размер *JA

Информация о матрице (обычный формат):

```
typedef struct
{
    int **strs;

    size_t m;
    size_t n;
} matrix_t;
```

Информация о матрице (обычный формат) представляет собой объединение на языке Си, состоящее из:

****strs** - Указатель на память ,содержащую указатели на строки

m - Кол-во строк матрицы

n - Кол-во столбцов матрицы

Информация о векторе (формат вектор-строки):

```
typedef struct
{
    int *B;
    size_t *JB;

    size_t size;
    size_t full_size;
} vector_str_t;
```

Информация о векторе (формат вектор-строки) представляет собой структуру на языке Си, состоящую из:

***B** - Указатель на память, содержащую значения ненулевых элементов

***JB** - Указатель на память, содержащую индексы ненулевых элементов

size - Кол-во ненулевых элементов

full_size - Кол-во элементов во всём векторе(в том числе и нулевых элементов)

Информация о векторе (обычный формат):

```
typedef struct
{
    int *coords;

    size_t full_size;
} vector_t;
```

Информация о векторе (обычный формат) представляет собой структуру на языке Си, состоящую из:

***coords** - Указатель на память, содержащую все элементы вектора

full_size - Кол-во элементов во всём векторе(в том числе и нулевых элементов)

Описание алгоритма

Для начала считывается код действия. Далее в зависимости от выбранного действия:

<init> (Инициализация матрицы и вектора):

Выбирается тип инициализации:

<autoinit>

- Вводятся размеры матрицы
- Вводится процент заполнения матрицы
- Программа заполняет матрицу

```
int matrix_autoinit(matrix_t *matrix, size_t m, size_t n, int percent);
```

(или

```
int vector_autoinit(vector_t *vector, size_t m, int percent);
```

- Программа конвертирует матрицу в другой формат (сохраняя изначальную матрицу)

```
int matrix_to_sparse(matrix_t *matrix, sparse_t *sparse);
```

(или

```
int vector_to_vector_str(vector_t *vector, vector_str_t *vector_str);
```

)

- Вводится процент заполнения вектора
- Программа заполняет вектор

```
int vector_autoinit(vector_t *vector, size_t m, int percent);
```

- Программа конвертирует вектор в другой формат (сохраняя изначальный вектор)

```
int vector_to_vector_str(vector_t *vector, vector_str_t *vector_str);
```

<self_init>

- Вводятся размеры матрицы
- Вводятся элементы матрицы

```
int matrix_init_manual(matrix_t *matrix, size_t m, size_t n);
```

(или

```
int vector_init_manual(vector_t *vector, size_t m);
```

)

- Программа конвертирует матрицу в другой формат (сохраняя изначальную матрицу)
- Вводятся элементы вектора

```
int vector_init_manual(vector_t *vector, size_t m);
```

- Программа конвертирует вектор в другой формат (сохраняя изначальный вектор)

<self_autoinit>

- Вводятся размеры матрицы
- Вводится процент заполнения матрицы
- Вводятся элементы матрицы

```
int matrix_autoinit_by_user(matrix_t *matrix, size_t m, size_t n, int percent);
```

(или

```
int vector_autoinit_by_user(vector_t *vector, size_t m, int percent);
```

)

- Программа конвертирует матрицу в другой формат (сохраняя изначальную матрицу)
- Вводится процент заполнения вектора
- Вводятся элементы вектора

```
int vector_autoinit_by_user(vector_t *vector, size_t m, int percent);
```

- Программа конвертирует вектор в другой формат (сохраняя изначальный вектор)

<output> <output_str_sparse> <output_usual> (Вывод матрицы и вектора)

```
void sparse_output(sparse_t *matrix, vector_str_t *vector);
```

```
void sparse_output_usual(sparse_t *matrix, vector_str_t *vector);
```

```
void vector_str_output(vector_str_t *vector);
```

```
void vector_str_output_usual(vector_str_t *vector);
```

- По порядку выводятся данные об элементах матрицы
- По порядку выводятся данные об элементах вектора

<count vector_str>

```
int vector_str_sparse_multiplic(vector_str_t *res, vector_str_t *vector, sparse_t *matrix);  
int vector_str_vector_str_multiplic(vector_str_t *res, vector_str_t *vector, vector_str_t *matrix_vector);
```

- Программа выделяет память для вектора, в котором будет содержаться результат
- Программа производит умножение, перенося результаты в вектор с результатом

<count vector>

```
int vector_matrix_multiplic(vector_t *res, vector_t *vector, matrix_t *matrix);  
int vector_vector_multiplic(vector_t *res, vector_t *vector, vector_t *matrix_vector);
```

- Программа выделяет память для вектора, в котором будет содержаться результат
- Программа производит умножение, перенося результаты в вектор с результатом

<statistics>

- Вводятся размеры матрицы
- Программа измеряет время умножения, кол-во выделенной памяти
- Программа выводит строку таблицы с информацией
- Данная процедура повторяется от 1% до 100%

<exit>

```
case CODE_EXIT:  
  
    all_free(&vector, &vector_str, &matrix, &sparse, &matrix_vector_str, &matrix_vector);  
  
    vector_str_free(&result_vector_str);  
    vector_free(&result_vector);  
  
    flag = false;  
  
    break;
```

- Освобождается память, выделенная под матрицы и векторы
- Программа завершает свою работу

Набор тестов

В ходе выполнения лабораторной работы были написаны тесты для проверки работы программы

Позитивные тесты

Номер теста	Входные данные	Ожидаемые выходные данные
01 - <init> <autoinit> Обычный тест	1 4 5 20 100	Матрицы (2 формы) и векторы (2 формы) с заполненностью 20% и 100% соответственно
02 - <init> <autoinit> Кол-во строк матрицы равно 1	1 1 5 20 100	Векторы, представляющую матрицу (2 формы), векторы (2 формы) с заполненностью 20% и 100% соответственно
03 - <init> <self_init> Обычный тест	1 4 5 1 2 3 4 5 1 2 3 4 5 0 1 2 0 0 0 0 0 0 1 1 2 3 4	Матрицы (2 формы) и векторы (2 формы) Матрица (в обычной форме): 1 2 3 4 5 1 2 3 4 5 0 1 2 0 0 0 0 0 0 1 Вектор в (обычной форме): 1 2 3 4
04 - <init> <self_init> Кол-во строк матрицы равно 1	1 1 5 0 1 2 0 0 1	Векторы, представляющую матрицу (2 формы), векторы (2 формы) Вектор-матрица (обычная форма): 0 1 2 0 0 Вектор (обычная форма): 1

05 - <init> <self_autoinit> Обычный тест	1 4 5 25 1 2 3 4 5 1 1	Матрицы (2 формы) и векторы (2 формы) с заполненностью 25% и 1% соответственно
06 - <init> <self_init> Кол-во строк матрицы равно 1	1 1 5 20 10053 1 1	Векторы, представляющую матрицу (2 формы), векторы (2 формы) с заполненностью 20% и 1% соответственно
07 - <output> <output_str_sparse> Обычный тест	1 4 5 20 100 2 1	Матрица (в разреженном формате) и вектор (в формате вектор-строки) с заполненностью 20% и 100% соответственно, выведенные на экран.
08 - <output> <output_str_sparse> Кол-во строк матрицы равно 1	1 1 5 20 100 2 1	Вектор, представляющий матрицу (в формате вектор-строки), вектор (в формате вектор-строки) с заполненностью 20% и 100% соответственно, выведенные на экран
09 - <output> <output_usual> Обычный тест	1 4 5 20 100 2 2	Матрица (в обычном формате) и вектор (в обычном формате) с заполненностью 20% и 100% соответственно, выведенные на экран.
10 - <output> <output_usual> Кол-во строк матрицы равно 1	1 1 5 20 100	Вектор, представляющий матрицу (в обычном формате), вектор (в обычном формате) с заполненностью 20% и

	2 2	100% соответственно, выведенные на экран
11 - <count vector_str> Умножение матрицы и вектора	1 4 5 20 100 3	Вектор в обычном формате, являющаяся результатом умножения матрицы в разреженном виде и вектор-строки
12 - <count vector_str> Умножение матрицы в форме вектора и вектора	1 1 5 20 100 3	Вектор в обычном формате, являющаяся результатом умножения матрицы в разреженном виде и вектор-строки
13 - <count vector> Умножение матрицы и вектора	1 4 5 20 100 4	Вектор в обычном формате, являющийся результатом умножения матрицы и вектора в обычном формате
14 - <count vector> Умножение матрицы в форме вектора и вектора	1 1 5 20 100 4	Вектор в обычном формате, являющийся результатом умножения матрицы и вектора в обычном формате
15 - <statistics> Вывод статистики по умножениям матрицы и вектора	5 4 5	Вывод таблицы с результатами умножения матрицы и вектора
16 - <statistics> Вывод статистики по умножениям матрицы в форме вектора и вектора	5 1 5	Вывод таблицы с результатами умножения матрицы в форме вектора и вектора
17 - <exit> Выход из программы	6	-

Негативные тесты

Номер теста	Входные данные	Выходные данные
01 - Код неправильный	7	INVALID CODE
02 - Код с иными символами	1.1	INVALID CODE
03 - <init> <autoinit> Введены неправильные размеры	1 0 1	INVALID COUNT
04 - <init> <self_autoinit> Введён неправильный процент	1 3 101	ERR PERCENT!
05 - <init> <self_autoinit> Введён неправильный элемент	1 3 100 1 2 3 y 4	ENTERED INVALID DATA
06 - <output> Данные отсутствуют	2	NO DATA
07 - <count vector_str> Данные отсутствуют	3	NO DATA
08 - <count vector> Данные отсутствуют	4	NO DATA
09 - <statistics> Указаны неверные размеры	5 5e 1	INVALID COUNT

Таблицы с результатами измерений времени и памяти при различных используемых форматах хранения и алгоритмах обработки матриц в их различном процентном заполнении нулями.

Ниже представлены результаты статистики при различных размерах матрицы

2 x 10000

STATISTICS (time in nsec) (memory in bytes)						
%percent	time (str_sparse)	time (vector_matrix)	mem (str_sparse)	mem (vector_matrix)	smaller memory	best time
1	136716	95289	82492	80064	vector_matrix	vector_matrix
2	292145	190725	84892	80064	vector_matrix	vector_matrix
3	432739	285963	87292	80064	vector_matrix	vector_matrix
4	588323	380979	89692	80064	vector_matrix	vector_matrix
5	742296	476213	92092	80064	vector_matrix	vector_matrix
6	956460	576734	94492	80064	vector_matrix	vector_matrix
7	1128919	672157	96892	80064	vector_matrix	vector_matrix
8	1373427	813490	99292	80064	vector_matrix	vector_matrix
9	1653617	954889	101692	80064	vector_matrix	vector_matrix
10	1893254	1058727	104092	80064	vector_matrix	vector_matrix
11	2210094	2058877	106492	80064	vector_matrix	vector_matrix
12	2472553	2191728	108892	80064	vector_matrix	vector_matrix
13	2758908	2348467	111292	80064	vector_matrix	vector_matrix
14	3139767	2498659	113692	80064	vector_matrix	vector_matrix
15	3458137	2642684	116092	80064	vector_matrix	vector_matrix
16	3757488	2730746	118492	80064	vector_matrix	vector_matrix
17	4042954	2825492	120892	80064	vector_matrix	vector_matrix
18	4267014	2898239	123292	80064	vector_matrix	vector_matrix
19	4492017	2965197	125692	80064	vector_matrix	vector_matrix
20	4716669	3050869	128092	80064	vector_matrix	vector_matrix
21	4964227	3140689	130492	80064	vector_matrix	vector_matrix
22	5218297	3236733	132892	80064	vector_matrix	vector_matrix
23	5494328	3357877	135292	80064	vector_matrix	vector_matrix
24	5901149	3454673	137692	80064	vector_matrix	vector_matrix
25	6192814	3525551	140092	80064	vector_matrix	vector_matrix
26	6466468	3601140	142492	80064	vector_matrix	vector_matrix
27	6777153	3684069	144892	80064	vector_matrix	vector_matrix
28	7055299	3753527	147292	80064	vector_matrix	vector_matrix
29	7370423	3867416	149692	80064	vector_matrix	vector_matrix
30	7702981	3964747	152092	80064	vector_matrix	vector_matrix
31	8013544	4030860	154492	80064	vector_matrix	vector_matrix
32	8316869	4123603	156892	80064	vector_matrix	vector_matrix
33	8596247	4186422	159292	80064	vector_matrix	vector_matrix
34	8861245	4248623	161692	80064	vector_matrix	vector_matrix
35	9142856	4310940	164092	80064	vector_matrix	vector_matrix
36	9411847	4372750	166492	80064	vector_matrix	vector_matrix
37	9771145	4463034	168892	80064	vector_matrix	vector_matrix
38	10067296	4523221	171292	80064	vector_matrix	vector_matrix
39	10376687	4615088	173692	80064	vector_matrix	vector_matrix
40	10698133	4697968	176092	80064	vector_matrix	vector_matrix
41	11123361	4765119	178492	80064	vector_matrix	vector_matrix
42	11491253	4923355	180892	80064	vector_matrix	vector_matrix
43	11784125	4985320	183292	80064	vector_matrix	vector_matrix
44	12062233	5047362	185692	80064	vector_matrix	vector_matrix
45	12361148	5109448	188092	80064	vector_matrix	vector_matrix
46	12682865	5172252	190492	80064	vector_matrix	vector_matrix
47	13006504	5246767	192892	80064	vector_matrix	vector_matrix
48	13280702	5308555	195292	80064	vector_matrix	vector_matrix
49	13545436	5368204	197692	80064	vector_matrix	vector_matrix
50	13808385	5427739	200092	80064	vector_matrix	vector_matrix

5 x 10000

STATISTICS (time in nsec) (memory in bytes)						
%percent	time (str_sparse)	time (vector_matrix)	mem (str_sparse)	mem (vector_matrix)	smaller memory	best time
1	181885	263091	86092	200100	str_sparse	str_sparse
2	305009	501461	92092	200100	str_sparse	str_sparse
3	467697	783025	98092	200100	str_sparse	str_sparse
4	652347	1055544	104092	200100	str_sparse	str_sparse
5	843818	1293620	110092	200100	str_sparse	str_sparse
6	1032132	1534066	116092	200100	str_sparse	str_sparse
7	1261616	1773109	122092	200100	str_sparse	str_sparse
8	1711069	2116788	128092	200100	str_sparse	str_sparse
9	1986164	2355475	134092	200100	str_sparse	str_sparse
10	2255763	2593783	140092	200100	str_sparse	str_sparse
11	2588562	2862293	146092	200100	str_sparse	str_sparse
12	2950589	3089723	152092	200100	str_sparse	str_sparse
13	3224730	3274558	158092	200100	str_sparse	str_sparse
14	3469516	3439591	164092	200100	str_sparse	vector_matrix
15	3731663	3626571	170092	200100	str_sparse	vector_matrix
16	4018534	3860010	176092	200100	str_sparse	vector_matrix
17	4328336	4060095	182092	200100	str_sparse	vector_matrix
18	4578674	4215488	188092	200100	str_sparse	vector_matrix
19	4835433	4366213	194092	200100	str_sparse	vector_matrix
20	5127889	4537142	200092	200100	str_sparse	vector_matrix
21	5428832	4691238	206092	200100	vector_matrix	vector_matrix
22	5789587	4879697	212092	200100	vector_matrix	vector_matrix
23	6071879	5029768	218092	200100	vector_matrix	vector_matrix
24	6339469	5179702	224092	200100	vector_matrix	vector_matrix
25	7744469	5825904	230092	200100	vector_matrix	vector_matrix
26	8312305	6047484	236092	200100	vector_matrix	vector_matrix
27	8616391	6207448	242092	200100	vector_matrix	vector_matrix
28	9022377	6472145	248092	200100	vector_matrix	vector_matrix
29	9451245	6711178	254092	200100	vector_matrix	vector_matrix
30	9763934	6866914	260092	200100	vector_matrix	vector_matrix
31	10082891	7039362	266092	200100	vector_matrix	vector_matrix
32	10399725	7194647	272092	200100	vector_matrix	vector_matrix
33	10705053	7389085	278092	200100	vector_matrix	vector_matrix
34	11471111	7760579	284092	200100	vector_matrix	vector_matrix
35	11815223	7921372	290092	200100	vector_matrix	vector_matrix
36	12161733	8080662	296092	200100	vector_matrix	vector_matrix
37	12533867	8305484	302092	200100	vector_matrix	vector_matrix
38	12917316	8532670	308092	200100	vector_matrix	vector_matrix
39	13261342	8687137	314092	200100	vector_matrix	vector_matrix
40	13730901	8842243	320104	200100	vector_matrix	vector_matrix
41	14422788	9024735	326104	200100	vector_matrix	vector_matrix
42	14879279	9180989	332104	200100	vector_matrix	vector_matrix
43	15308276	9331010	338104	200100	vector_matrix	vector_matrix
44	15753092	9537278	344104	200100	vector_matrix	vector_matrix
45	16270866	9698449	350104	200100	vector_matrix	vector_matrix
46	16702748	9849691	356104	200100	vector_matrix	vector_matrix
47	17152291	10003356	362104	200100	vector_matrix	vector_matrix
48	17599153	10243694	368104	200100	vector_matrix	vector_matrix
49	18051874	10400307	374104	200100	vector_matrix	vector_matrix
50	18490436	10556028	380104	200100	vector_matrix	vector_matrix

50 x 10000

STATISTICS (time in nsec) (memory in bytes)						
%percent	time (str_sparse)	time (vector_matrix)	mem (str_sparse)	mem (vector_matrix)	smaller memory	best time
1	303133	2407955	140092	2000640	str_sparse	str_sparse
2	604787	4686878	200092	2000640	str_sparse	str_sparse
3	867007	6156766	260092	2000640	str_sparse	str_sparse
4	1304179	7792023	320104	2000640	str_sparse	str_sparse
5	1687735	9316621	380104	2000640	str_sparse	str_sparse
6	2334743	12325463	440116	2000640	str_sparse	str_sparse
7	3033369	13960270	500116	2000640	str_sparse	str_sparse
8	3879357	17870765	560128	2000640	str_sparse	str_sparse
9	5046496	20938555	620128	2000640	str_sparse	str_sparse
10	6990948	22553333	680140	2000640	str_sparse	str_sparse
11	8440762	24576658	740140	2000640	str_sparse	str_sparse
12	10027837	26129089	800152	2000640	str_sparse	str_sparse
13	12013541	27704344	860152	2000640	str_sparse	str_sparse
14	13933414	30665307	920164	2000640	str_sparse	str_sparse
15	15755478	32117830	980164	2000640	str_sparse	str_sparse
16	18359380	33786677	1040176	2000640	str_sparse	str_sparse
17	20803704	36297638	1100176	2000640	str_sparse	str_sparse
18	23995897	38291022	1160188	2000640	str_sparse	str_sparse
19	26801191	40154021	1220188	2000640	str_sparse	str_sparse
20	29848354	41897455	1280200	2000640	str_sparse	str_sparse
21	32925257	43701624	1340200	2000640	str_sparse	str_sparse
22	36290490	45432556	1400212	2000640	str_sparse	str_sparse
23	39787192	46971013	1460212	2000640	str_sparse	str_sparse
24	43613383	48418228	1520224	2000640	str_sparse	str_sparse
25	47680139	49946913	1580224	2000640	str_sparse	str_sparse
26	52282477	51494686	1640236	2000640	str_sparse	vector_matrix
27	56848507	52983357	1700236	2000640	str_sparse	vector_matrix
28	62679232	54725363	1760248	2000640	str_sparse	vector_matrix
29	68057383	57280398	1820248	2000640	str_sparse	vector_matrix
30	75579278	58909167	1880260	2000640	str_sparse	vector_matrix
31	81935377	60780938	1940260	2000640	str_sparse	vector_matrix
32	88978543	62361063	2000272	2000640	str_sparse	vector_matrix
33	96488283	63863683	2060272	2000640	vector_matrix	vector_matrix
34	104010931	65735553	2120284	2000640	vector_matrix	vector_matrix
35	111656064	67331157	2180284	2000640	vector_matrix	vector_matrix
36	120003510	68907157	2240296	2000640	vector_matrix	vector_matrix
37	128745699	70845813	2300296	2000640	vector_matrix	vector_matrix
38	140190610	72412307	2360308	2000640	vector_matrix	vector_matrix
39	152195581	73982717	2420308	2000640	vector_matrix	vector_matrix
40	162530566	75802788	2480320	2000640	vector_matrix	vector_matrix
41	172814933	77557348	2540320	2000640	vector_matrix	vector_matrix
42	183030131	79502283	2600332	2000640	vector_matrix	vector_matrix
43	192906225	81042468	2660332	2000640	vector_matrix	vector_matrix
44	203651753	82574104	2720344	2000640	vector_matrix	vector_matrix
45	215875829	84224232	2780344	2000640	vector_matrix	vector_matrix
46	229126243	85824215	2840356	2000640	vector_matrix	vector_matrix
47	240404708	87434739	2900356	2000640	vector_matrix	vector_matrix
48	255743638	89009704	2960368	2000640	vector_matrix	vector_matrix
49	268187571	90639111	3020368	2000640	vector_matrix	vector_matrix

100 x 1000

STATISTICS (time in nsec) (memory in bytes)						
%percent	time (str_sparse)	time (vector_matrix)	mem (str_sparse)	mem (vector_matrix)	smaller memory	best time
1	34467	457652	20092	401240	str_sparse	str_sparse
2	90734	914111	32104	401240	str_sparse	str_sparse
3	181034	1378535	44116	401240	str_sparse	str_sparse
4	311006	1848484	56128	401240	str_sparse	str_sparse
5	485599	2309291	68140	401240	str_sparse	str_sparse
6	716243	2773039	80152	401240	str_sparse	str_sparse
7	1000164	3226783	92164	401240	str_sparse	str_sparse
8	1340251	3689003	104176	401240	str_sparse	str_sparse
9	1731542	4154048	116188	401240	str_sparse	str_sparse
10	2248919	4671054	128200	401240	str_sparse	str_sparse
11	2755308	5139266	140212	401240	str_sparse	str_sparse
12	3348073	5658301	152224	401240	str_sparse	str_sparse
13	4010653	6120225	164236	401240	str_sparse	str_sparse
14	5201217	6838997	176248	401240	str_sparse	str_sparse
15	6540548	7590298	188260	401240	str_sparse	str_sparse
16	8186359	8459804	200272	401240	str_sparse	str_sparse
17	9977571	9161608	212284	401240	str_sparse	vector_matrix
18	11889045	9860884	224296	401240	str_sparse	vector_matrix
19	14059867	10579352	236308	401240	str_sparse	vector_matrix
20	16478912	11286092	248320	401240	str_sparse	vector_matrix
21	19412145	12073733	260332	401240	str_sparse	vector_matrix
22	22117957	12915018	272344	401240	str_sparse	vector_matrix
23	25118937	13554791	284356	401240	str_sparse	vector_matrix
24	27203779	14032945	296368	401240	str_sparse	vector_matrix
25	29491871	14512945	308380	401240	str_sparse	vector_matrix
26	32022637	15024297	320392	401240	str_sparse	vector_matrix
27	34563757	15597508	332404	401240	str_sparse	vector_matrix
28	37282535	16102572	344416	401240	str_sparse	vector_matrix
29	41589745	16838472	356428	401240	str_sparse	vector_matrix
30	46301973	17511876	368440	401240	str_sparse	vector_matrix
31	52271279	18309205	380452	401240	str_sparse	vector_matrix
32	57602016	19205664	392464	401240	str_sparse	vector_matrix
33	63280949	19957227	404476	401240	vector_matrix	vector_matrix
34	67570440	21144735	416488	401240	vector_matrix	vector_matrix
35	73443961	21925783	428500	401240	vector_matrix	vector_matrix
36	79471438	22784488	440512	401240	vector_matrix	vector_matrix
37	86362885	23533281	452524	401240	vector_matrix	vector_matrix
38	93583483	24251535	464536	401240	vector_matrix	vector_matrix
39	98412136	24746850	476548	401240	vector_matrix	vector_matrix
40	103878135	25399596	488560	401240	vector_matrix	vector_matrix
41	112133184	26214988	500572	401240	vector_matrix	vector_matrix
42	118974336	26740911	512584	401240	vector_matrix	vector_matrix
43	125571527	27630828	524596	401240	vector_matrix	vector_matrix
44	134560425	28400995	536608	401240	vector_matrix	vector_matrix
45	143685108	29131591	548620	401240	vector_matrix	vector_matrix
46	153858989	29904154	560632	401240	vector_matrix	vector_matrix
47	163739747	30883116	572644	401240	vector_matrix	vector_matrix
48	175087316	31711976	584656	401240	vector_matrix	vector_matrix
49	185732753	32565064	596668	401240	vector_matrix	vector_matrix

50 x 50

STATISTICS (time in nsec) (memory in bytes)						
%percent	time (str_sparse)	time (vector_matrix)	mem (str_sparse)	mem (vector_matrix)	smaller memory	best time
1	3623	15617	792	10640	str_sparse	str_sparse
2	6857	31888	1092	10640	str_sparse	str_sparse
3	10820	48173	1392	10640	str_sparse	str_sparse
4	16907	67582	1704	10640	str_sparse	str_sparse
5	22694	82837	2004	10640	str_sparse	str_sparse
6	29444	94051	2316	10640	str_sparse	str_sparse
7	36861	109924	2616	10640	str_sparse	str_sparse
8	46517	121096	2928	10640	str_sparse	str_sparse
9	59141	139018	3228	10640	str_sparse	str_sparse
10	70242	150204	3540	10640	str_sparse	str_sparse
11	86058	167838	3840	10640	str_sparse	str_sparse
12	99641	178954	4152	10640	str_sparse	str_sparse
13	113188	190178	4452	10640	str_sparse	str_sparse
14	133444	210141	4764	10640	str_sparse	str_sparse
15	149190	222336	5064	10640	str_sparse	str_sparse
16	173680	233752	5376	10640	str_sparse	str_sparse
17	198386	247049	5676	10640	str_sparse	str_sparse
18	227665	264625	5988	10640	str_sparse	str_sparse
19	257645	275835	6288	10640	str_sparse	str_sparse
20	281267	286994	6600	10640	str_sparse	str_sparse
21	314467	303030	6900	10640	str_sparse	vector_matrix
22	351168	320012	7212	10640	str_sparse	vector_matrix
23	392362	336681	7512	10640	str_sparse	vector_matrix
24	448464	352164	7824	10640	str_sparse	vector_matrix
25	494220	368118	8124	10640	str_sparse	vector_matrix
26	545396	385138	8436	10640	str_sparse	vector_matrix
27	598450	402233	8736	10640	str_sparse	vector_matrix
28	657277	418593	9048	10640	str_sparse	vector_matrix
29	718391	435352	9348	10640	str_sparse	vector_matrix
30	785454	452471	9660	10640	str_sparse	vector_matrix
31	837869	463694	9960	10640	str_sparse	vector_matrix
32	914067	479758	10272	10640	str_sparse	vector_matrix
33	988012	493225	10572	10640	str_sparse	vector_matrix
34	1061152	507767	10884	10640	vector_matrix	vector_matrix
35	1118473	519138	11184	10640	vector_matrix	vector_matrix
36	1204803	535466	11496	10640	vector_matrix	vector_matrix
37	1295508	551757	11796	10640	vector_matrix	vector_matrix
38	1374097	568025	12108	10640	vector_matrix	vector_matrix
39	1449129	584238	12408	10640	vector_matrix	vector_matrix
40	1535293	595677	12720	10640	vector_matrix	vector_matrix
41	1639637	628527	13020	10640	vector_matrix	vector_matrix
42	2125911	669622	13332	10640	vector_matrix	vector_matrix
43	3108554	693069	13632	10640	vector_matrix	vector_matrix
44	3661988	718111	13944	10640	vector_matrix	vector_matrix
45	3839595	739269	14244	10640	vector_matrix	vector_matrix
46	4006438	762257	14556	10640	vector_matrix	vector_matrix
47	4158080	780757	14856	10640	vector_matrix	vector_matrix
48	4327895	804205	15168	10640	vector_matrix	vector_matrix
49	4545860	822047	15468	10640	vector_matrix	vector_matrix
50	4729535	843979	15780	10640	vector_matrix	vector_matrix

100 x 100

STATISTICS (time in nsec) (memory in bytes)						
%percent	time (str_sparse)	time (vector_matrix)	mem (str_sparse)	mem (vector_matrix)	smaller memory	best time
1	5043	40289	2092	41240	str_sparse	str_sparse
2	12273	84162	3304	41240	str_sparse	str_sparse
3	22864	128634	4516	41240	str_sparse	str_sparse
4	36612	170809	5728	41240	str_sparse	str_sparse
5	56266	214683	6940	41240	str_sparse	str_sparse
6	81562	258826	8152	41240	str_sparse	str_sparse
7	109694	303566	9364	41240	str_sparse	str_sparse
8	144249	348201	10576	41240	str_sparse	str_sparse
9	184404	392412	11788	41240	str_sparse	str_sparse
10	230082	436722	13000	41240	str_sparse	str_sparse
11	282092	481575	14212	41240	str_sparse	str_sparse
12	342084	528386	15424	41240	str_sparse	str_sparse
13	385018	556408	16636	41240	str_sparse	str_sparse
14	444766	584177	17848	41240	str_sparse	str_sparse
15	501727	613444	19060	41240	str_sparse	str_sparse
16	564412	642093	20272	41240	str_sparse	str_sparse
17	676292	687059	21484	41240	str_sparse	str_sparse
18	799458	733066	22696	41240	str_sparse	vector_matrix
19	933131	780303	23908	41240	str_sparse	vector_matrix
20	1078583	825005	25120	41240	str_sparse	vector_matrix
21	1237267	871682	26332	41240	str_sparse	vector_matrix
22	1410740	1088729	27544	41240	str_sparse	vector_matrix
23	1599421	1133204	28756	41240	str_sparse	vector_matrix
24	1800191	1180744	29968	41240	str_sparse	vector_matrix
25	2028294	1239615	31180	41240	str_sparse	vector_matrix
26	2254008	1285518	32392	41240	str_sparse	vector_matrix
27	2491993	1329974	33604	41240	str_sparse	vector_matrix
28	2749266	1374440	34816	41240	str_sparse	vector_matrix
29	3057736	1421409	36028	41240	str_sparse	vector_matrix
30	3353112	1466894	37240	41240	str_sparse	vector_matrix
31	3658919	1513403	38452	41240	str_sparse	vector_matrix
32	3960659	1557656	39664	41240	str_sparse	vector_matrix
33	4296066	1627645	40876	41240	str_sparse	vector_matrix
34	4856021	1702361	42088	41240	vector_matrix	vector_matrix
35	5179487	1741597	43300	41240	vector_matrix	vector_matrix
36	5487769	1777152	44512	41240	vector_matrix	vector_matrix
37	5880097	1845320	45724	41240	vector_matrix	vector_matrix
38	6238244	1876959	46936	41240	vector_matrix	vector_matrix
39	6532288	1906534	48148	41240	vector_matrix	vector_matrix
40	6859612	1935094	49360	41240	vector_matrix	vector_matrix
41	7158609	1962727	50572	41240	vector_matrix	vector_matrix
42	7506301	1995724	51784	41240	vector_matrix	vector_matrix
43	7836282	2024185	52996	41240	vector_matrix	vector_matrix
44	8185150	2052629	54208	41240	vector_matrix	vector_matrix
45	8547688	2118758	55420	41240	vector_matrix	vector_matrix
46	8905955	2146949	56632	41240	vector_matrix	vector_matrix
47	9280759	2179532	57844	41240	vector_matrix	vector_matrix
48	9662998	2207103	59056	41240	vector_matrix	vector_matrix
49	10062477	2234812	60268	41240	vector_matrix	vector_matrix
50	10489284	2265958	61480	41240	vector_matrix	vector_matrix

1000 x 1000

STATISTICS (time in nsec) (memory in bytes)						
%percent	time (str_sparse)	time (vector_matrix)	mem (str_sparse)	mem (vector_matrix)	smaller memory	best time
1	445748	6099088	128200	4012040	str_sparse	str_sparse
2	1527309	12897786	248320	4012040	str_sparse	str_sparse
3	3507794	18868357	368440	4012040	str_sparse	str_sparse
4	8264606	27878196	488560	4012040	str_sparse	str_sparse
5	14592000	35205588	608680	4012040	str_sparse	str_sparse
6	22082896	42229965	728800	4012040	str_sparse	str_sparse
7	31352331	49124222	848920	4012040	str_sparse	str_sparse
8	44297103	56233685	969040	4012040	str_sparse	str_sparse
9	61344145	63355078	1089160	4012040	str_sparse	str_sparse
10	82169561	70668470	1209280	4012040	str_sparse	vector_matrix
11	107080861	77010408	1329400	4012040	str_sparse	vector_matrix
12	138647169	83218866	1449520	4012040	str_sparse	vector_matrix
13	173942563	88820300	1569640	4012040	str_sparse	vector_matrix
14	213068567	94014371	1689760	4012040	str_sparse	vector_matrix
15	258469399	101034269	1809880	4012040	str_sparse	vector_matrix
16	310799775	107336850	1930000	4012040	str_sparse	vector_matrix
17	367783331	112587344	2050120	4012040	str_sparse	vector_matrix
18	430343099	119114365	2170240	4012040	str_sparse	vector_matrix
19	497385434	124010972	2290360	4012040	str_sparse	vector_matrix
20	585503313	132309058	2410480	4012040	str_sparse	vector_matrix
21	665030533	137781532	2530600	4012040	str_sparse	vector_matrix
22	757003676	143529414	2650720	4012040	str_sparse	vector_matrix
23	861494150	149306730	2770840	4012040	str_sparse	vector_matrix
24	963313597	154750811	2890960	4012040	str_sparse	vector_matrix
25	1080527988	159943939	3011080	4012040	str_sparse	vector_matrix
26	1199585868	165673659	3131200	4012040	str_sparse	vector_matrix
27	1327467136	171749145	3251320	4012040	str_sparse	vector_matrix
28	1467040152	177491327	3371440	4012040	str_sparse	vector_matrix
29	1614439675	182845568	3491560	4012040	str_sparse	vector_matrix
30	1768347715	188294063	3611680	4012040	str_sparse	vector_matrix
31	1936225339	194548497	3731800	4012040	str_sparse	vector_matrix
32	2118594211	200437102	3851920	4012040	str_sparse	vector_matrix
33	2307261425	206831480	3972040	4012040	str_sparse	vector_matrix
34	2495374253	213359165	4092160	4012040	vector_matrix	vector_matrix
35	2698186504	219453638	4212280	4012040	vector_matrix	vector_matrix
36	2914735106	225136677	4332400	4012040	vector_matrix	vector_matrix
37	3136996067	230810717	4452520	4012040	vector_matrix	vector_matrix
38	3366163626	237164341	4572640	4012040	vector_matrix	vector_matrix
39	3616374101	242863844	4692760	4012040	vector_matrix	vector_matrix
40	3871677383	248384038	4812880	4012040	vector_matrix	vector_matrix
41	4141972189	253865899	4933000	4012040	vector_matrix	vector_matrix
42	4428012237	259710246	5053120	4012040	vector_matrix	vector_matrix
43	4719283220	265185535	5173240	4012040	vector_matrix	vector_matrix
44	5018080624	271754803	5293360	4012040	vector_matrix	vector_matrix
45	5325299429	276801225	5413480	4012040	vector_matrix	vector_matrix
46	5646923688	282988283	5533600	4012040	vector_matrix	vector_matrix
47	5979863438	288594082	5653720	4012040	vector_matrix	vector_matrix
48	6329051814	295624513	5773840	4012040	vector_matrix	vector_matrix
49	6706309173	301445810	5893960	4012040	vector_matrix	vector_matrix
50	7085068385	307359080	6014080	4012040	vector_matrix	vector_matrix

1000 x 100

STATISTICS (time in nsec) (memory in bytes)						
%percent	time (str_sparse)	time (vector_matrix)	mem (str_sparse)	mem (vector_matrix)	smaller memory	best time
1	47940	474797	13000	412040	str_sparse	str_sparse
2	196246	949041	25120	412040	str_sparse	str_sparse
3	498226	1422696	37240	412040	str_sparse	str_sparse
4	1134633	1897752	49360	412040	str_sparse	str_sparse
5	2039701	2387464	61480	412040	str_sparse	str_sparse
6	3445836	2866785	73600	412040	str_sparse	vector_matrix
7	4848751	3232477	85720	412040	str_sparse	vector_matrix
8	6216396	3802692	97840	412040	str_sparse	vector_matrix
9	7943388	4138290	109960	412040	str_sparse	vector_matrix
10	10394591	4522813	122080	412040	str_sparse	vector_matrix
11	12780889	4833792	134200	412040	str_sparse	vector_matrix
12	16243054	5306706	146320	412040	str_sparse	vector_matrix
13	19468579	5649068	158440	412040	str_sparse	vector_matrix
14	23727690	5992757	170560	412040	str_sparse	vector_matrix
15	29167119	6455166	182680	412040	str_sparse	vector_matrix
16	35646823	6869038	194800	412040	str_sparse	vector_matrix
17	42734855	7287072	206920	412040	str_sparse	vector_matrix
18	49133027	7620375	219040	412040	str_sparse	vector_matrix
19	58178224	7989808	231160	412040	str_sparse	vector_matrix
20	66432172	8331941	243280	412040	str_sparse	vector_matrix
21	75282339	8794713	255400	412040	str_sparse	vector_matrix
22	84855903	9154044	267520	412040	str_sparse	vector_matrix
23	95884965	9533087	279640	412040	str_sparse	vector_matrix
24	106554711	9919960	291760	412040	str_sparse	vector_matrix
25	120641216	10347988	303880	412040	str_sparse	vector_matrix
26	132370542	10771343	316000	412040	str_sparse	vector_matrix
27	145087165	11125328	328120	412040	str_sparse	vector_matrix
28	161332669	11502578	340240	412040	str_sparse	vector_matrix
29	176782586	11921299	352360	412040	str_sparse	vector_matrix
30	193162869	12269868	364480	412040	str_sparse	vector_matrix
31	209567048	12639334	376600	412040	str_sparse	vector_matrix
32	228402675	13256462	388720	412040	str_sparse	vector_matrix
33	247002137	13630608	400840	412040	str_sparse	vector_matrix
34	266077684	14145474	412960	412040	vector_matrix	vector_matrix
35	286637064	14515499	425080	412040	vector_matrix	vector_matrix
36	308565509	14921851	437200	412040	vector_matrix	vector_matrix
37	331606936	15296598	449320	412040	vector_matrix	vector_matrix
38	366521934	16061737	461440	412040	vector_matrix	vector_matrix
39	393611382	16451042	473560	412040	vector_matrix	vector_matrix
40	419790985	16854051	485680	412040	vector_matrix	vector_matrix
41	449400691	17234893	497800	412040	vector_matrix	vector_matrix
42	481680553	17669707	509920	412040	vector_matrix	vector_matrix
43	512409575	18047197	522040	412040	vector_matrix	vector_matrix
44	545415518	18515366	534160	412040	vector_matrix	vector_matrix
45	579798307	18886884	546280	412040	vector_matrix	vector_matrix
46	613876337	19288436	558400	412040	vector_matrix	vector_matrix
47	648417742	19651942	570520	412040	vector_matrix	vector_matrix
48	685155901	20055608	582640	412040	vector_matrix	vector_matrix
49	722697016	20392345	594760	412040	vector_matrix	vector_matrix
50	763232210	20877982	606880	412040	vector_matrix	vector_matrix

10000 x 50

STATISTICS (time in nsec) (memory in bytes)						
%percent	time (str_sparse)	time (vector_matrix)	mem (str_sparse)	mem (vector_matrix)	smaller memory	best time
1	26078	231762	6600	212040	str_sparse	str_sparse
2	119959	463702	12720	212040	str_sparse	str_sparse
3	271793	699872	18840	212040	str_sparse	str_sparse
4	580641	936784	24960	212040	str_sparse	str_sparse
5	1246404	1288441	31080	212040	str_sparse	str_sparse
6	2208976	1527050	37200	212040	str_sparse	vector_matrix
7	3726666	1954292	43320	212040	str_sparse	vector_matrix
8	4857520	2255636	49440	212040	str_sparse	vector_matrix
9	7107285	2751309	55560	212040	str_sparse	vector_matrix
10	8851529	3007782	61680	212040	str_sparse	vector_matrix
11	10763951	3245813	67800	212040	str_sparse	vector_matrix
12	13606776	3486848	73920	212040	str_sparse	vector_matrix
13	16153129	3749022	80040	212040	str_sparse	vector_matrix
14	20279834	4148445	86160	212040	str_sparse	vector_matrix
15	25076309	4537158	92280	212040	str_sparse	vector_matrix
16	30058574	4865897	98400	212040	str_sparse	vector_matrix
17	36515536	5197618	104520	212040	str_sparse	vector_matrix
18	42256165	5482979	110640	212040	str_sparse	vector_matrix
19	48067257	5757179	116760	212040	str_sparse	vector_matrix
20	54392871	6033410	122880	212040	str_sparse	vector_matrix
21	61208448	6334183	129000	212040	str_sparse	vector_matrix
22	70872651	6663117	135120	212040	str_sparse	vector_matrix
23	81922699	7033138	141240	212040	str_sparse	vector_matrix
24	90705118	7291876	147360	212040	str_sparse	vector_matrix
25	100626457	7760036	153480	212040	str_sparse	vector_matrix
26	118656604	8183976	159600	212040	str_sparse	vector_matrix
27	137505695	8814384	165720	212040	str_sparse	vector_matrix
28	156814246	9325359	171840	212040	str_sparse	vector_matrix
29	176077711	9711801	177960	212040	str_sparse	vector_matrix
30	191976813	9993304	184080	212040	str_sparse	vector_matrix
31	211934260	10352305	190200	212040	str_sparse	vector_matrix
32	231139166	10655713	196320	212040	str_sparse	vector_matrix
33	248797225	10988326	202440	212040	str_sparse	vector_matrix
34	274151075	11427304	208560	212040	str_sparse	vector_matrix
35	298628308	11832490	214680	212040	vector_matrix	vector_matrix
36	323537003	12209843	220800	212040	vector_matrix	vector_matrix
37	347197144	12743934	226920	212040	vector_matrix	vector_matrix
38	379362187	13588244	233040	212040	vector_matrix	vector_matrix
39	413155556	13987120	239160	212040	vector_matrix	vector_matrix
40	447738779	14432406	245280	212040	vector_matrix	vector_matrix
41	481416245	14822601	251400	212040	vector_matrix	vector_matrix
42	509665191	15266119	257520	212040	vector_matrix	vector_matrix
43	536306697	15542719	263640	212040	vector_matrix	vector_matrix
44	567705649	16022974	269760	212040	vector_matrix	vector_matrix
45	599011572	16555328	275880	212040	vector_matrix	vector_matrix
46	636471425	16945258	282000	212040	vector_matrix	vector_matrix
47	670218227	17265624	288120	212040	vector_matrix	vector_matrix
48	711418862	17666839	294240	212040	vector_matrix	vector_matrix
49	756246977	18225047	300360	212040	vector_matrix	vector_matrix

10000 x 10

STATISTICS (time in nsec) (memory in bytes)							
%percent	time (str_sparse)	time (vector_matrix)	mem (str_sparse)	mem (vector_matrix)	smaller memory	best time	
1	442082	478501	13360	520040	str_sparse	str_sparse	
2	1669647	929925	26560	520040	str_sparse	vector_matrix	
3	8898897	1581439	39760	520040	str_sparse	vector_matrix	
4	19004915	2344731	52960	520040	str_sparse	vector_matrix	
5	36255068	3234089	66160	520040	str_sparse	vector_matrix	
6	58202461	4270469	79360	520040	str_sparse	vector_matrix	
7	79772973	4822112	92560	520040	str_sparse	vector_matrix	
8	105222309	5364029	105760	520040	str_sparse	vector_matrix	
9	134176522	6112459	118960	520040	str_sparse	vector_matrix	
10	179043893	6773395	132160	520040	str_sparse	vector_matrix	
11	230735095	7624468	145360	520040	str_sparse	vector_matrix	
12	290463285	8343891	158560	520040	str_sparse	vector_matrix	
13	358467077	8883170	171760	520040	str_sparse	vector_matrix	
14	424202442	9359337	184960	520040	str_sparse	vector_matrix	
15	497848947	9935403	198160	520040	str_sparse	vector_matrix	
16	576789329	10439593	211360	520040	str_sparse	vector_matrix	
17	679830030	10962737	224560	520040	str_sparse	vector_matrix	
18	781102214	11507686	237760	520040	str_sparse	vector_matrix	
19	902098667	11971452	250960	520040	str_sparse	vector_matrix	
20	1025946880	12466614	264160	520040	str_sparse	vector_matrix	
21	1142892280	12800367	277360	520040	str_sparse	vector_matrix	
22	1228723674	13214847	290560	520040	str_sparse	vector_matrix	
23	1326759797	13590026	303760	520040	str_sparse	vector_matrix	
24	1440164752	13932928	316960	520040	str_sparse	vector_matrix	
25	1553081103	14303082	330160	520040	str_sparse	vector_matrix	
26	1668621256	14722596	343360	520040	str_sparse	vector_matrix	
27	1793281650	15099890	356560	520040	str_sparse	vector_matrix	
28	1945021723	15431394	369760	520040	str_sparse	vector_matrix	
29	2094219434	15786673	382960	520040	str_sparse	vector_matrix	
30	2244717983	16170245	396160	520040	str_sparse	vector_matrix	
31	2407314241	16556936	409360	520040	str_sparse	vector_matrix	
32	2587988240	16922911	422560	520040	str_sparse	vector_matrix	
33	2770906780	17320582	435760	520040	str_sparse	vector_matrix	
34	2963793391	17707378	448960	520040	str_sparse	vector_matrix	
35	3164768665	18076409	462160	520040	str_sparse	vector_matrix	
36	3381528581	18461392	475360	520040	str_sparse	vector_matrix	
37	3605149439	18821384	488560	520040	str_sparse	vector_matrix	
38	3842656386	19203166	501760	520040	str_sparse	vector_matrix	
39	4089562636	19644889	514960	520040	str_sparse	vector_matrix	
40	4340796479	20015533	528160	520040	vector_matrix	vector_matrix	
41	4616352956	20328846	541360	520040	vector_matrix	vector_matrix	
42	4906137151	20723334	554560	520040	vector_matrix	vector_matrix	
43	5188342566	21096150	567760	520040	vector_matrix	vector_matrix	
44	5488570014	21481242	580960	520040	vector_matrix	vector_matrix	
45	5797467778	21957128	594160	520040	vector_matrix	vector_matrix	
46	6127663693	22279675	607360	520040	vector_matrix	vector_matrix	
47	6464086739	22645960	620560	520040	vector_matrix	vector_matrix	
48	6816835142	22949294	633760	520040	vector_matrix	vector_matrix	
49	7183869774	23412620	646960	520040	vector_matrix	vector_matrix	
50	7563376592	23819201	660160	520040	vector_matrix	vector_matrix	

10000 x 5

STATISTICS (time in nsec) (memory in bytes)						
%percent	time (str_sparse)	time (vector_matrix)	mem (str_sparse)	mem (vector_matrix)	smaller memory	best time
1	220324	274349	7320	320040	str_sparse	str_sparse
2	1072845	540180	14520	320040	str_sparse	vector_matrix
3	4111138	998957	21720	320040	str_sparse	vector_matrix
4	6919711	1210852	28920	320040	str_sparse	vector_matrix
5	12249874	1629272	36120	320040	str_sparse	vector_matrix
6	20285058	1977946	43320	320040	str_sparse	vector_matrix
7	35139127	2254556	50520	320040	str_sparse	vector_matrix
8	47278363	2552243	57720	320040	str_sparse	vector_matrix
9	62540323	3097607	64920	320040	str_sparse	vector_matrix
10	83927071	3536844	72120	320040	str_sparse	vector_matrix
11	106914685	3946009	79320	320040	str_sparse	vector_matrix
12	136344307	4410182	86520	320040	str_sparse	vector_matrix
13	168512754	4704638	93720	320040	str_sparse	vector_matrix
14	205788892	5340361	100920	320040	str_sparse	vector_matrix
15	248155556	5711008	108120	320040	str_sparse	vector_matrix
16	303571057	6108874	115320	320040	str_sparse	vector_matrix
17	356285668	6434822	122520	320040	str_sparse	vector_matrix
18	418545041	6791436	129720	320040	str_sparse	vector_matrix
19	489085898	7143926	136920	320040	str_sparse	vector_matrix
20	561978960	7436745	144120	320040	str_sparse	vector_matrix
21	630114462	7680134	151320	320040	str_sparse	vector_matrix
22	709295091	8013807	158520	320040	str_sparse	vector_matrix
23	797275108	8264515	165720	320040	str_sparse	vector_matrix
24	880726945	8518087	172920	320040	str_sparse	vector_matrix
25	978962078	8779434	180120	320040	str_sparse	vector_matrix
26	1068166374	8966025	187320	320040	str_sparse	vector_matrix
27	1138193103	9152094	194520	320040	str_sparse	vector_matrix
28	1223075100	9389906	201720	320040	str_sparse	vector_matrix
29	1300420212	9578946	208920	320040	str_sparse	vector_matrix
30	1383529605	9819813	216120	320040	str_sparse	vector_matrix
31	1467203516	10027083	223320	320040	str_sparse	vector_matrix
32	1561431946	10217292	230520	320040	str_sparse	vector_matrix
33	1654704668	10401208	237720	320040	str_sparse	vector_matrix
34	1758626114	10566274	244920	320040	str_sparse	vector_matrix
35	1864756372	10750649	252120	320040	str_sparse	vector_matrix
36	1990851486	10950074	259320	320040	str_sparse	vector_matrix
37	2108252564	11145396	266520	320040	str_sparse	vector_matrix
38	2231321383	11376253	273720	320040	str_sparse	vector_matrix
39	2371481736	11591419	280920	320040	str_sparse	vector_matrix
40	2508646600	11775767	288120	320040	str_sparse	vector_matrix
41	2641936443	12041086	295320	320040	str_sparse	vector_matrix
42	2789724613	12251583	302520	320040	str_sparse	vector_matrix
43	2940154655	12566513	309720	320040	str_sparse	vector_matrix
44	3097555914	12710821	316920	320040	str_sparse	vector_matrix
45	3251743999	12893104	324120	320040	vector_matrix	vector_matrix
46	3411751462	13084388	331320	320040	vector_matrix	vector_matrix
47	3578215530	13297476	338520	320040	vector_matrix	vector_matrix
48	3761837548	13496512	345720	320040	vector_matrix	vector_matrix
49	3941965182	13662336	352920	320040	vector_matrix	vector_matrix
50	4124870076	13852429	360120	320040	vector_matrix	vector_matrix

На основе полученных данных можно сделать таблицу результатов исследования

Размер матриц	С какого процента вычисления str_sparse (str_str) перестает быть выгодной по памяти (в %)	С какого процента вычисления str_sparse (str_str) перестает быть выгодной по времени (в %)
2 x 10000	-	-
5 x 10000	21	14
50 x 10000	33	26
100 x 1000	33	17
50 x 50	34	21
100 x 100	34	18
1000 x 1000	34	10
1000 x 100	34	6
10000 x 50	35	6
10000 x 10	40	2
10000 x 5	45	2

На основе полученных данных можно сделать следующие выводы:

- Если строк меньше, чем столбцов, вычисления с использованием вектор-строки и матрицы в разреженном формате являются невыгодными и по памяти, и по времени при любых процентах заполнения
- При увеличении кол-ва строк при неизменности кол-ва столбцов вычисления с использованием вектор-строки и матрицы в разреженном формате становятся выгодными по памяти до 33% заполнения (стремятся к этому проценту) и выгодными по времени до 17%-26% заполнения.
- При квадратной матрице вычисления с использованием вектор-строки и матрицы в разреженном формате становятся выгодными по памяти до ~33%, но при увеличении размерности максимальный процент выгоды по времени уменьшается
- При большем кол-ве строк и меньшем кол-ве столбцов максимальный процент выгоды по памяти вычислений с использованием вектор-строки и матрицы в разреженном формате увеличивается, но уменьшается выгода по времени

Выводы по проделанной работе

Если необходимо производить вычисления с матрицами/векторами, большая часть элементов которых составляют нулевые элементы, стоит представлять матрицы и векторы в особом разреженном формате. По мере возрастания кол-ва строк матрицы (если перед нами вектор-строка) максимальный процент разреженности, при котором памяти требуется меньше, чем для хранения в обычном формате, возрастает. По мере возрастания кол-ва столбцов максимальный процент разреженности, при котором существует эффективность по времени по сравнению с хранением в обычном формате, возрастает. Если же мы работаем с квадратной матрицей, хранение в разреженном формате эффективно по памяти примерно до 33% заполненности включительно, но при увеличении размеров, максимальный процент заполненности, при котором требуется меньше времени для выполнения вычислений, постепенно уменьшается. По мере увеличения размеров матрицы эффективность по времени падает.

Контрольные вопросы

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

Разреженная матрица - матрица, в которой преобладают нулевые значения.

Схемы хранения: схема Кнута, схема Рейнболта-Местиньи, схема Лакрума, схема Чанга и Густавсона

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

Для хранения разреженной матрицы выделяется память только под информацию о ненулевых элементах. Для хранения обычной памяти выделяется память под ВСЕ элементы матрицы

3. Каков принцип обработки разреженной матрицы?

Алгоритмы обработки разреженных матриц предусматривают действия только с ненулевыми элементами.

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

В случае, когда матрицу нельзя назвать разреженной (большая часть элементов ненулевые), при небольшом размере матрицы. Эффективность применение тех или иных алгоритмов обработки матриц зависят от того, сколько памяти мы готовы затратить и насколько быстро мы хотим провести обработку.