

Задание №1

Умножение матриц

1 Матрицы общего вида

Постановка задачи:

Даны произвольные квадратные матрицы вещественных чисел A и B (размера $n \times n$ каждая). Требуется вычислить матрицу $C = A \cdot B$. Известно, что n достаточно велико ($n \sim 10^4$). Постарайтесь минимизировать время выполнения программы (любыми возможными способами).

Представление данных и результатов:

Матрицы A и B заданы в текстовых файлах *data1.dat* и *data2.dat* одинаковой структуры. В первой строке файла содержится символ #, пробел и натуральное число (размер матрицы n). Остальные n строк содержат соответствующие строки матрицы, элементы одной строки разделены пробелами (одним или несколькими).

Полученную в результате матрицу C требуется вывести в файл *result.dat* с такой же структурой. Собственно умножение матриц должно быть вынесено в отдельную функцию или процедуру, к которой обращается программа, реализующая ввод-вывод данных.

Можно считать, что объем памяти, необходимый для хранения элементов матриц A , B , C , не превосходит объема доступной оперативной памяти.

2 Трехдиагональные матрицы

Постановка задачи:

Даны квадратные матрицы вещественных чисел A и B (размера $n \times n$ каждая) следующего вида:

$$A = \begin{pmatrix} a_{11} & a_{21} & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ a_{12} & a_{22} & a_{32} & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & a_{23} & a_{33} & a_{43} & 0 & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots & a_{n-2,n-2} & a_{n-2,n-1} & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & a_{n-1,n-2} & a_{n-1,n-1} & a_{n-1,n} \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & a_{n,n-1} & a_{n,n} \end{pmatrix}.$$

Такие матрицы принято называть трехдиагональными, они нередко встречаются в вычислительных задачах.

Требуется вычислить матрицу $C = A \cdot B$. Известно, что n достаточно велико ($n \sim 10^7$), что, в частности, означает невозможность хранения матриц в оперативной памяти «как есть», со всеми заведомо нулевыми элементами.

Представление данных и результатов:

Матрицы A и B заданы в текстовых файлах *data1.dat* и *data2.dat* одинаковой структуры. В первой строке файла содержится символ #, пробел и натуральное число (размер матрицы n). Остальные n строк содержат лежащие на трех центральных

диагоналях элементы соответствующих строк матрицы, элементы одной строки разделены пробелами (одним или несколькими). Например, для $n = 4$ файл может иметь следующий вид:

```
# 4
2.350  9.045
4.678 -3.567  0.567
-9.765 -7.456  10.92
1.145  5.782
```

Полученную в результате пятидиагональную матрицу C требуется вывести в файл *result.dat* с аналогичной структурой (строки со второй и далее имеют длину 3–4–5–5–...–5–5–4–3). Собственно умножение матриц должно быть вынесено в отдельную функцию или процедуру, к которой обращается программа, реализующая ввод-вывод данных.

Можно считать, что объем памяти, необходимый для хранения *ненулевых* элементов матриц A , B , C , не превосходит объема доступной оперативной памяти.