

**Правительство Российской Федерации**

**Федеральное государственное автономное образовательное**

**учреждение высшего образования «Национальный**

**исследовательский университет «Высшая школа экономики»**

Факультет компьютерных наук

Департамент программной инженерии

**Отчет к домашнему заданию по дисциплине**

**«Архитектура вычислительных систем»**

Работу выполнил:

Студент группы БПИ-195 Гуницкий Р.Я.

**Москва 2020**

## СОДЕРЖАНИЕ

1. ТЕКСТ ЗАДАНИЯ .....	4
2. ПРИМЕНЯЕМЫЕ РАСЧЕТНЫЕ МЕТОДЫ .....	4
3. ТЕСТИРОВАНИЕ ПРОГРАММЫ .....	5
ИСТОЧНИКИ .....	8
ПРИЛОЖЕНИЕ 1 .....	9
КОД ПРОГРАММЫ .....	9

## 1. ТЕКСТ ЗАДАНИЯ

Определить ранг матрицы. Входные данные: целое положительное число  $n$ , произвольная матрица  $A$  размерности  $n \times n$ . Количество потоков является входным параметром, при этом размерность матриц может быть не кратна количеству потоков.

## 2. ПРИМЕНЯЕМЫЕ РАСЧЕТНЫЕ МЕТОДЫ

Для вычисления ранга матрицы, матрица приводится к ступенчатому виду методом Гауса, после чего подсчитывается количество ее ненулевых строк. В программе реализована парадигма итеративного параллелизма. Данный метод был выбран для ускорения вычислений новых строк матрицы путем реализации алгоритма Гаусса.

На вход программе подается размерность матрицы и количество потоков, которые будут работать над задачей.

Программа работает следующим образом: считывается размер матрицы и количество потоков, которые будут приводить ее к ступенчатому виду, после чего запускается цикл в котором находится первая строка с минимальным индексом ведущего элемента. Далее выбираются строки индекс ведущего элемента которых равен индексу ведущего элемента выбранной строки и из элементов этих строк вычитаются элементы выбранной строки, умноженные на ведущий элемент строки из которой производится вычитание и поделенный на ведущий элемент выбранной строки. Таким образом в итоге получается ступенчатая матрица и количество ее ненулевых строк равно ее рангу.

### 3. ТЕСТИРОВАНИЕ ПРОГРАММЫ

```

C:\> Консоль отладки Microsoft Visual Studio
Введите размер матрицы [1, 20]:5
42      76      11      75      69
50      25      57      14      84
34      8       18      49      47
21      76      92      6       24
25      93      53      94      24
Введите количество потоков:5
Приведенная к ступенчатому виду матрица:
42      0       0       0       0
0       -65.48  0       0       0
0       0       -26.78  0       0
0       0       0       -75.19  0
0       0       0       0       -16.29
Ранг матрицы: 5

```

Рисунок 1 – Нахождение ранга матрицы 5 на 5 с 5 потоками

```

Введите размер матрицы [1, 20]:20
65      86      3       85      58      88      0       29      28      13      26      46      89      91      63      33      20      1       53      34
67      94      45      56      95      90      3       18      70      72      94      13      24      48      79      70      93      32      62      3
62      11      33      38      2       63      87      50      11      91      4       64      53      24      52      425      58      9       55      8
58      1       4       8       73      81      62      0       95      40      61      66      56      76      0       91      16      15      6       16
11      98      94      45      46      88      2       39      37      95      27      5       77      85      1       52      53      51      4       51
37      17      69      48      54      38      48      28      72      5       31      44      88      36      39      49      30      72      85      50
97      2       11      16      88      92      8       66      95      51      90      13      83      99      94      41      77      19      34      18
2       83      98      98      86      63      87      56      67      62      39      76      0       87      20      410      51      97      58
68      67      98      63      83      51      28      67      85      98      72      94      25      84      83      94      24      50      86      14
70      69      90      43      22      2       66      52      32      85      56      20      51      23      78      83      84      0       82      72
0       0       52      58      57      65      17      36      75      46      52      70      14      34      73      97      88      44      13      54
0       78      89      90      61      24      76      93      20      66      83      62      58      28      52      86      0       9       15      47
62      42      29      49      24      92      39      39      34      88      57      64      51      10      48      97      58      53      38      15
16      33      77      95      48      30      26      56      88      92      66      4       78      10      29      39      27      39      94      85
64      23      71      15      66      22      37      64      73      34      7       87      10      91      15      90      63      21      46
81      54      15      36      62      34      96      88      97      0       21      75      20      76      69      35      52      85      72      41
14      7       58      31      42      4       33      92      72      66      16      73      48      60      54      24      24      39      38      64
70      62      52      0       16      59      84      27      26      20      53      19      94      57      5       53      8       80      27      73
7       19      25      25      76      73      91      7       80      0       0       52      72      37      71      74      44      24      76      97
67      9       29      48      1       43      19      6       18      1       13      65      47      51      95      58      58      81      15      39
Введите количество потоков:7
Приведенная к ступенчатому виду матрица:
65      0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
0       5.98      0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
0       0       233.29      0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
0       0       0       -46.50      0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
0       0       0       0       131      0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       -108.35      0       0       0       0       0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       -116.33      0       0       0       0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       55.32      0       0       0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       10.07      0       0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       138.31      0       0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0       15.41      0       0       0       0       0       0       0       0       0
0       0       0       0       0       0       0       0       0       0       0       126.82      0       0       0       0       88      0       0       0
0       0       0       0       0       0       0       0       0       0       0       0       622.75      0       0       88      0       0       0       0
0       0       0       0       0       0       0       0       0       0       0       0       0       -58.68      0       88      0       0       0       0
0       0       0       0       0       0       0       0       0       0       0       0       0       0       154.36      68      0       0       0       0
0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       -6.21      8       0       0       0
0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0-273      0       0       0
0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       260.49      0       0
0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       -203.38      0
0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       0       -100.85
Ранг матрицы: 20

```

Рисунок 2 - Нахождение ранга матрицы 20 на 20 с 7 потоками

```

Консоль отладки Microsoft Visual Studio
Введите размер матрицы [1, 20]:10
60 16 33 87 15 51 2 14 46 63
70 78 47 27 28 73 7 24 94 26
18 73 7 58 29 30 49 19 79 82
29 28 51 65 78 32 36 74 83 10
34 85 73 68 64 60 93 26 55 31
77 70 14 22 85 42 91 98 80 98
49 66 24 78 73 16 86 97 74 26
41 23 72 2 42 28 72 78 16 35
62 40 49 34 55 65 40 14 74 22
89 54 21 39 58 27 55 84 31 54
Введите количество потоков:7
Приведенная к ступенчатому виду матрица:
60 0 0 0 0 0 0 0 0 0
0 59.33 0 0 0 0 0 0 0 0
0 0 -12.67 0 0 0 0 0 0 0
0 0 0 346.66 0 0 0 0 0 0
0 0 0 0 -62.5 0 0 0 0 0
0 0 0 0 0 -11.89 0 0 0 0
0 0 0 0 0 0 -218.16 0 0 0
0 0 0 0 0 0 0 -6.92 0 0
0 0 0 0 0 0 0 0 28.26 0
0 0 0 0 0 0 0 0 0 1038.79
Ранг матрицы: 10

```

Рисунок 3 – Нахождение ранга матрицы 10 на 10 с 7 потоками

```

C:\Users\admin\source\repos\ConsoleApplication1\Debug\ConsoleApplication1.exe
Введите размер матрицы [1, 20]:13
52 12 75 99 82 84 29 45 89 75 44 77 20
20 46 12 37 68 2 58 4 44 45 4 68 38
1 45 39 85 45 12 14 64 76 96 53 65 1
17 71 74 22 70 70 76 95 80 60 44 10 38
81 83 53 98 85 0 27 47 36 52 91 86 52
85 81 90 46 11 90 35 25 43 22 67 63 16
68 4 28 80 90 76 76 22 53 9 32 45 15
99 81 10 4 17 16 13 62 20 33 58 12 74
41 53 20 69 13 50 11 38 70 7 6 97 38
47 46 62 36 20 81 74 78 10 67 18 53 38
10 67 35 42 86 71 13 94 8 23 48 64 41
93 81 7 77 68 89 8 68 69 82 89 97 88
43 99 62 96 4 63 43 51 14 95 2 81 92
Введите количество потоков:10
Приведенная к ступенчатому виду матрица:
52 0 0 0 0 0 0 0 0 0 0 0 0
0 41.38 0 0 0 0 0 0 0 0 0 0 0
0 0 55.79 0 0 0 0 0 0 0 0 0 0
0 0 0 -124.63 0 0 0 0 0 0 0 0 0
0 0 0 0 -97.11 0 0 0 0 0 0 0 0
0 0 0 0 0 65.37 0 0 0 0 0 0 0
0 0 0 0 0 0 -2.55 0 0 0 0 0 0
0 0 0 0 0 0 0 169.91 0 0 0 0 0
0 0 0 0 0 0 0 0 -7.37 0 0 0 0
0 0 0 0 0 0 0 0 0 567.48 0 0 0
0 0 0 0 0 0 0 0 0 0 72.53 0 0
0 0 0 0 0 0 0 0 0 0 0 66.04 0
0 0 0 0 0 0 0 0 0 0 0 0 47.23
Ранг матрицы: 13

```

Рисунок 4 – Нахождение ранга матрицы 13 на 13 с 10 потоками

```

C:\Users\admin\source\repos\ConsoleApplication1\Debug
Введите размер матрицы [1, 20]:5
54      14      73      29      63
34      62      17      10      33
13      55      21      11      26
95      40      37      0       52
0       0       0       0       0
Введите количество потоков:3
Приведенная к ступенчатому виду матрица:
54      0       0       0       29.56
0       44.38   0       0       2.85
0       0       67.33   0       6.19
0       0       0       -51.02  -58.83
0       0       0       0       0
Ранг матрицы: 4

```

Рисунок 5 – Нахождение ранга матрицы 5 на 5 с одной нулевой строкой с 3 потоками

```

Консоль отладки Microsoft Visual Studio
Введите размер матрицы [1, 20]:-2
Введенны некорректные данные . . .
Попробуйте ввести число еще раз:21
Введенны некорректные данные . . .
Попробуйте ввести число еще раз:6
11      81      24      57      3       95
70      69      23      31      20      9
80      26      18      38      94      54
43      80      44      77      9       84
9       13      15      39      0       23
18      15      24      19      64      98
Введите количество потоков:20
Введенны некорректные данные . . .
Попробуйте ввести число еще раз:-9
Введенны некорректные данные . . .
Попробуйте ввести число еще раз:3
Приведенная к ступенчатому виду матрица:
11      0       0       0       0       0
0       -446.45  0       0       0       0
0       0       7.07   0       0       0
0       0       0       77      0       0
0       0       0       0       -4.56   0
0       0       0       0       0       -187.6
Ранг матрицы: 6

```

Рисунок 6 – ввод некорректных данных

**ИСТОЧНИКИ**

1. SoftCraft, сайт по учебной дисциплине. [Электронный ресурс] <http://softcraft.ru/>
2. Парадигмы параллельного программирования. [Электронный ресурс]  
<http://www.williamspublishing.com/PDF/5-8459-0388-2/part.pdf>
3. Метод Гаусса. [Электронный ресурс]  
[https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4\\_%D0%93%D0%B0%D1%83%D1%81%D1%81%D0%B0](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%93%D0%B0%D1%83%D1%81%D1%81%D0%B0)

## КОД ПРОГРАММЫ

```

#include <iostream>
#include<vector>
#include<thread>
#include<mutex>
#include<string>
#include<ctime>
#include<cmath>

std::mutex mtx1;

/// <summary>
/// Выводит матрицу в консоль
/// </summary>
/// <param name="matrix"></param>
/// <param name="size"></param>
void printMatrix(double** matrix, int size) {
    for (size_t i = 0; i < size; i++)
    {
        for (size_t j = 0; j < size; j++)
        {
            std::cout << matrix[i][j] << "\t";
        }
        std::cout << std::endl;
    }
}

/// <summary>
/// Создает случайную матрицу
/// </summary>
/// <param name="size"></param>
/// <returns></returns>
double** createMatrix(int size) {
    double** matrix = new double* [size];

    for (size_t i = 0; i < size; i++)
        matrix[i] = new double[size];

    for (size_t i = 0; i < size; i++)
        for (size_t j = 0; j < size; j++) {
            matrix[i][j] = rand() % 100;
        }

    return matrix;
}

/// <summary>
/// Удаляет матрицу
/// </summary>
/// <param name="matrix"></param>
/// <param name="sizeOfMatrix"></param>
void deleteMatrix(double** matrix, int sizeOfMatrix) {
    for (size_t i = 0; i < sizeOfMatrix; i++)
    {
        delete[] matrix[i];
    }
    delete[] matrix;
}

/// <summary>

```



```

/// Считывает число
/// </summary>
/// <param name="num"></param>
/// <param name="minValue"></param>
/// <param name="maxValue"></param>
void ReadNumber(int& num, int minValue, int maxValue = INT_MAX) {
    std::cin >> num;
    while (num < minValue || num > maxValue) {
        std::cout << "Введенны некорректные данные . . ." << std::endl;
        std::cout << "Попробуйте ввести число еще раз:";
        std::cin >> num;
    }
}

/// <summary>
/// Вычитает из элементов строк, индекс ведущего элемента которых равен
/// индексу ведущего элемента выбранной строки, элементы выбранной строки
/// умноженные на matrix[i][firstElemIndex] и деленные на matrix[lineInd][firstElemIndex]
/// </summary>
/// <param name="matrix">матрица</param>
/// <param name="size">размер матрицы</param>
/// <param name="lineInd">индекс выбранной строки</param>
/// <param name="elemInd">индекс элемента в этой строке</param>
/// <param name="firstElemIndex">индекс ведущего элемента выбранной строки</param>
void SubLines(double** matrix, int size, int lineInd, int elemInd, int firstElemIndex) {
    for (size_t i = 0; i < size; i++)
        if (i != lineInd && matrix[i][firstElemIndex] != 0)
        {
            mtx1.lock();
            double elem1 = matrix[lineInd][elemInd];
            double elem2 = matrix[i][firstElemIndex];
            double elem3 = matrix[lineInd][firstElemIndex];
            matrix[i][elemInd] -= floor((elem1 * elem2 / elem3) * 100.0 + 0.5) /
100.0;

            if (abs(matrix[i][elemInd]) < 0.00001)
                matrix[i][elemInd] = 0;
            mtx1.unlock();
        }
}

void ThreadsSubs(double** matrix, int size, int lineInd, int firstElemIndex, int
startInd, int endInd) {
    for (int i = endInd - 1; i >= startInd; i--) {
        SubLines(matrix, size, lineInd, i, firstElemIndex);
    }
}

/// <summary>
/// Находит индекс ведущего элемента строки
/// </summary>
/// <param name="line">строка</param>
/// <param name="size">размер матрицы</param>
/// <returns></returns>
int FindStartIndex(double* line, int size) {
    for (size_t i = 0; i < size; i++)
        if (line[i] != 0)
            return i;
    return -1;
}

/// <summary>
/// Находит ранг ступенчатой матрицы
/// </summary>
/// <param name="matrix"></param>

```

```

/// <param name="size"></param>
/// <returns></returns>
int Rg(double** matrix, int size) {
    int rg = size;
    for (int i = 0; i < size; ++i) {
        int countZeroes = 0;
        for (int j = 0; j < size; ++j) {
            if (matrix[i][j] == 0)
                countZeroes++;
        }
        if (countZeroes == size)
            rg--;
    }
    return rg;
}

int main()
{
    setlocale(LC_ALL, "Russian");
    srand(time(0));

    std::cout << "Введите размер матрицы [1, 20]:";
    int sizeOfMatrix;
    ReadNumber(sizeOfMatrix, 1, 20);
    double** matrix = createMatrix(sizeOfMatrix);
    printMatrix(matrix, sizeOfMatrix);

    std::cout << "Введите количество потоков:";
    int countThread;
    ReadNumber(countThread, 1, sizeOfMatrix);

    int maxIndex = -1;
    for (size_t i = 0; i < sizeOfMatrix; i++)
    {
        //Находим строку с минимальным индексом ведущего элемента
        //и сохраняем ее индекс и индекс ее ведущего элемента
        int elemInd = sizeOfMatrix + 1; //индекс ведущего элемента
        int lineInd = sizeOfMatrix + 1; //индекс строки
        for (size_t j = 0; j < sizeOfMatrix; j++)
        {
            int startInd = FindStartIndex(matrix[j], sizeOfMatrix);
            if (startInd > maxIndex && startInd < elemInd)
            {
                lineInd = j;
                elemInd = startInd;
                maxIndex = startInd;
            }
        }
        //Если индекс ведущего элемента не изменился, то в матрице больше нет
        //ненулевых строк, поэтому выходим из цикла
        if (elemInd == sizeOfMatrix + 1)
            break;

        if (countThread > sizeOfMatrix - elemInd)
            countThread = sizeOfMatrix - elemInd;

        //Разбиваем задачу на потоки
        int countColumnsForThread = sizeOfMatrix / countThread;
        std::thread* threads = new std::thread[countThread];
        //Каждый поток занимается своей частью матрицы
        for (size_t i = 0; i < countThread; i++)
        {
            int startInd = i * countColumnsForThread;

```

```

        int endInd = i < countThread - 1 ? (i + 1) * countThread :
sizeofMatrix;
        threads[i] = std::thread(ThreadsSubs, matrix, sizeofMatrix, lineInd,
elemInd, startInd, endInd);
    }

    //Объединяем потоки и удаляем их
    for (size_t i = 0; i < countThread; i++)
        threads[i].join();
    delete[] threads;
}
std::cout << "Приведенная к ступенчатому виду матрица:" << std::endl;
printMatrix(matrix, sizeofMatrix);
std::cout << "Ранг матрицы: " << Rg(matrix, sizeofMatrix) << std::endl;
deleteMatrix(matrix, sizeofMatrix);
}

```