

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук
Образовательная программа бакалавриата «Программная инженерия»

ДОМАШНЕЕ ЗАДАНИЕ №3

Нахождение алгебраических дополнений элементов матрицы

Вариант 3

Пояснительная записка

Работу выполнил
студент группы БПИ192
Ахметьянов Арслан Рашидович
17.11.2020

Москва 2020

Содержание

| | |
|--|-----------|
| Постановка задачи..... | 3 |
| Область допустимых входных данных | 3 |
| Выходные данные | 3 |
| Используемая модель вычислений | 4 |
| Примеры входных данных | 5 |
| Список источников..... | 9 |
| Приложение (исходный код программы) | 10 |

Постановка задачи

Согласно выданному заданию было необходимо реализовать следующий проект:

Найти определитель матрицы A . 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.

Область допустимых входных данных

Программа принимает на вход 2 целых числа, передаваемых как аргументы командной строки при запуске программы.

- Первый аргумент – количество потоков, дополнительно выделяемых для вычисления алгебраических дополнений. Ожидается целое положительное число.
- Второй аргумент – размерность матрицы. Ожидается целое положительное число, большее или равное двум. Данное ограничение связано с тем, что вычисление алгебраических дополнений для матрицы размерности 1 лишено практического смысла.

Матрица, для элементов которой вычисляются алгебраические дополнения, является целочисленной и генерируется случайно. Генерируемые значения – целые числа в отрезке $[-200; 200]$.

Выходные данные

В ходе выполнения программы выводится следующая информация:

1. Информация об ошибке в случае некорректных входных данных.
2. Подсчитанное алгебраическое дополнение для каждого элемента матрицы.

Результат представляется в виде строки следующего вида для каждого элемента:

Algebraical complement of element (%coord1, %coord2) = %res

Используемая модель вычислений

В приведенном решении задачи используются две модели вычислений.

1) Модель **Управляющие и рабочие**

В приведенном решении управляющим потоком является главный поток программы. Именно в нем происходит распределение элементов матрицы между потоками - каждый поток рассматривает все элементы матрицы, номер которых кратен номеру потока. Номер элемента определяется в порядке слева направо, затем сверху вниз. Каждый из потоков печатает результат обработки каждого из полученного элементов, основной же поток ждет выполнения всех порожденных им, а затем завершает работу программы.

2) Модель **Итеративного параллелизма**

Все создаваемые потоки работают над идентичными задачами – подсчет алгебраических дополнений элементов матрицы. Каждая пара потоков исполняет идентичную процедуру, которая не зависит от результатов исполнения данной процедуры другими потоками; вся совокупность потоков же решает общую задачу программы – нахождение алгебраических дополнений всех элементов матрицы.

Примеры входных данных

В приведенных примерах ввод – это аргументы командной строки для запуска программы. В качестве вывода приведен лишь один из возможных примеров ответа, поскольку матрицы генерируются случайно.

1. Ввод: 4, 4

Пример ожидаемого вывода:

```
Generated matrix is:
-59 -33 34 0
69 24 -22 58
62 -36 5 45
-19 -73 61 -9
Algebraical complement of element (0, 0) = -93760
Algebraical complement of element (1, 0) = 30636
Algebraical complement of element (2, 0) = -26392
Algebraical complement of element (3, 0) = 65472
Algebraical complement of element (0, 3) = 104767
Algebraical complement of element (0, 2) = -60287
Algebraical complement of element (1, 2) = 203130
Algebraical complement of element (2, 2) = -221189
Algebraical complement of element (0, 1) = -38890
Algebraical complement of element (1, 3) = 58830
Algebraical complement of element (3, 2) = 203115
Algebraical complement of element (1, 1) = 154512
Algebraical complement of element (2, 3) = 22273
Algebraical complement of element (3, 3) = -39003
Algebraical complement of element (2, 1) = -180706
Algebraical complement of element (3, 1) = 92214
```

2. Ввод: 2, 1

Пример ожидаемого вывода:

```
Matrix size should be 2 or greater! Got: 1.
```

3. Ввод: 1, 2

Пример ожидаемого вывода:

Generated matrix is:

-59 -33

34 0

Algebraical complement of element (0, 0) = 0

Algebraical complement of element (0, 1) = -34

Algebraical complement of element (1, 0) = 33

Algebraical complement of element (1, 1) = -59

4. Ввод: 18, 9

Пример ожидаемого вывода:

Generated matrix is:

-59 -33 34 0 69 24 -22 58 62

-36 5 45 -19 -73 61 -9 95 42

-73 -64 91 -96 2 53 -8 82 -79

16 18 -5 -53 26 71 38 -31 12

-33 -1 -65 -6 3 -89 22 33 -27

-36 41 11 -47 -32 47 -56 -38 57

-63 -41 23 41 29 78 16 -65 90

-58 -12 6 -60 42 -36 -52 -54 -95

-10 29 70 50 -94 1 93 48 -71

Algebraical complement of element (1, 2) = -568627245

Algebraical complement of element (0, 7) = 1408291739

Algebraical complement of element (0, 3) = 1286858308

Algebraical complement of element (0, 6) = 47736511

Algebraical complement of element (1, 6) = 737970031

Algebraical complement of element (0, 2) = 1038529731

Algebraical complement of element (1, 5) = -659841392

Algebraical complement of element (1, 0) = -288339359

Algebraical complement of element (1, 1) = 1998733919

Algebraical complement of element (1, 7) = -2057186389

Algebraical complement of element (0, 5) = 885353712

Algebraical complement of element (1, 8) = -153964222

Algebraical complement of element (1, 4) = -599554924

Algebraical complement of element (0, 1) = -25944241

Algebraical complement of element (0, 0) = -1095123375

Algebraical complement of element (0, 4) = 1448372308

Algebraical complement of element (1, 3) = -1752471772

Algebraical complement of element (0, 8) = 578380706

Algebraical complement of element (2, 7) = 1654022995

Algebraical complement of element (3, 2) = 252024691

Algebraical complement of element (2, 3) = -277432508
 Algebraical complement of element (2, 6) = -1442894473
 Algebraical complement of element (3, 6) = -2044691985
 Algebraical complement of element (3, 5) = -1475509968
 Algebraical complement of element (3, 0) = -1853836127
 Algebraical complement of element (3, 1) = 1215060735
 Algebraical complement of element (2, 2) = -56219141
 Algebraical complement of element (3, 4) = 397550132
 Algebraical complement of element (2, 0) = -649315335
 Algebraical complement of element (3, 8) = 1994918722
 Algebraical complement of element (2, 4) = -831427564
 Algebraical complement of element (2, 5) = -1073787440
 Algebraical complement of element (3, 7) = 626600619
 Algebraical complement of element (3, 3) = 572579140
 Algebraical complement of element (2, 1) = -1959829337
 Algebraical complement of element (2, 8) = 2015690418
 Algebraical complement of element (4, 7) = -706435635
 Algebraical complement of element (5, 2) = -345865159
 Algebraical complement of element (4, 3) = 584479900
 Algebraical complement of element (4, 6) = -141482647
 Algebraical complement of element (5, 6) = -57804819
 Algebraical complement of element (5, 5) = -1214527440
 Algebraical complement of element (5, 0) = -347683101
 Algebraical complement of element (5, 1) = -1320684035
 Algebraical complement of element (5, 4) = 1181840284
 Algebraical complement of element (4, 4) = -1483440084
 Algebraical complement of element (4, 2) = -1908169275
 Algebraical complement of element (4, 0) = 920536551
 Algebraical complement of element (5, 7) = 463489089
 Algebraical complement of element (4, 5) = -807166160
 Algebraical complement of element (5, 8) = 1943777670
 Algebraical complement of element (4, 8) = 1558371662
 Algebraical complement of element (4, 1) = -313990279
 Algebraical complement of element (5, 3) = 180196524
 Algebraical complement of element (6, 7) = -269348357
 Algebraical complement of element (6, 3) = -1201698012
 Algebraical complement of element (7, 2) = 636382584
 Algebraical complement of element (7, 6) = 1965619896
 Algebraical complement of element (7, 5) = -2084609792
 Algebraical complement of element (6, 6) = -1394200737
 Algebraical complement of element (7, 0) = -1549266936
 Algebraical complement of element (7, 4) = 675602432
 Algebraical complement of element (7, 1) = -472015272
 Algebraical complement of element (6, 2) = 517685251

Algebraical complement of element (6, 4) = -1887569708
 Algebraical complement of element (6, 0) = -178983471
 Algebraical complement of element (7, 8) = -1006225488
 Algebraical complement of element (6, 5) = -659213648
 Algebraical complement of element (7, 7) = -1818550376
 Algebraical complement of element (6, 1) = 108756591
 Algebraical complement of element (6, 8) = -1539074078
 Algebraical complement of element (7, 3) = 1770183808
 Algebraical complement of element (8, 7) = -2074170535
 Algebraical complement of element (8, 3) = 1967416012
 Algebraical complement of element (8, 6) = 1522150821
 Algebraical complement of element (8, 2) = -100010607
 Algebraical complement of element (8, 4) = 1630980444
 Algebraical complement of element (8, 0) = 75727147
 Algebraical complement of element (8, 5) = -289896144
 Algebraical complement of element (8, 1) = -557733163
 Algebraical complement of element (8, 8) = -1539868586

5. Ввод: 5, 3

Пример ожидаемого вывода:

Generated matrix is:

-59 -33 34

0 69 24

-22 58 62

Algebraical complement of element (0, 0) = 2886
 Algebraical complement of element (1, 2) = 4148
 Algebraical complement of element (0, 1) = -528
 Algebraical complement of element (1, 1) = -2910
 Algebraical complement of element (0, 2) = 1518
 Algebraical complement of element (2, 1) = 1416
 Algebraical complement of element (2, 0) = -3138
 Algebraical complement of element (1, 0) = 4018
 Algebraical complement of element (2, 2) = -4071

Список источников

1. <https://docplayer.ru/48706922-Lekciya-5-paradigmy-parallelnogo-programmirovaniya.html>
2. <http://softcraft.ru/edu/comparch/>
3. <http://softcraft.ru/edu/comparch/tasks/t03/>
4. http://www.k2x2.info/kompyutery_i_internet/parallelnoe_i_raspredelennoe_programmirovanie_na_s/p6.php#metkadoc25

Приложение (исходный код программы)

Вариант 3 (Ахметьянов Арслан Рашидович, БПИ 192)

Найти алгебраическое дополнение для каждого элемента 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.

```
#include <iostream>
#include <thread>
#include <vector>
#include <mutex>
#include <string>
#include <algorithm>

/* Matrix class. */
struct Matrix {
public:

    /* Empty matrix constructor. */
    Matrix(int n) {
        matrix_size_ = n;
        matrix_ = new short*[n];
        for (int i = 0; i < n; i++)
        {
            matrix_[i] = new short[n];
        }
    }

    /* Matrix constructor. Recieves array to initialize values. */
    Matrix(int n, int* array) {
        matrix_size_ = n;
        int pos = 0;
        matrix_ = new short* [n];
        for (int i = 0; i < n; i++)
        {
            matrix_[i] = new short[n];
            for (int j = 0; j < n; j++) {
                matrix_[i][j] = array[pos++];
            }
        }
    }

    /* Matrix destructor. */
    ~Matrix() {
        delete[] matrix_;
    }

    /* Calculates matrix determinant. */
    int getDeterminant() {
        if (this->matrix_size_ == 1) {
            return matrix_[0][0];
        }
    }
}
```

```

        if (this->matrix_size_ == 2) {
            return matrix_[0][0] * matrix_[1][1] -
                matrix_[1][0] * matrix_[0][1];
        }
        int res = 0;
        for (int i = 0; i < this->matrix_size_; ++i) {
            int sign = i % 2 == 0 ? 1 : -1;
            res += sign * matrix_[0][i] * buildSubmatrix(0, i)-
>getDeterminant();
        }
        return res;
    }

    /* Prints matrix to the console. */
    void printMatrix() {
        for (int i = 0; i < matrix_size_; ++i) {
            for (int j = 0; j < matrix_size_; ++j) {
                std::cout << matrix_[i][j] << " ";
            }
            std::cout << "\n";
        }
    }

    /* Builds submatrix without row and column with given numbers. */
    Matrix* buildSubmatrix(int excluded_row_num, int excluded_column_num) {
        Matrix* res = new Matrix(this->matrix_size_ - 1);
        for (int i = 0; i < this->matrix_size_ - 1; ++i) {
            int row = i < excluded_row_num ? i : i + 1;
            for (int j = 0; j < this->matrix_size_ - 1; ++j) {
                int column = j < excluded_column_num ? j : j + 1;
                res->matrix_[i][j] = this->matrix_[row][column];
            }
        }
        return res;
    }

private:
    short** matrix_;

    int matrix_size_;
};

std::mutex mutex;

/* Calculates algebraical complement for the given matrix element. */
void getAlgebraicalComplement(Matrix& matrix, int row, int column) {
    int sign = (row + column) % 2 == 0 ? 1 : -1;
    int res = sign * matrix.buildSubmatrix(row, column)->getDeterminant();
    mutex.lock();
    std::cout << "Algebraical complement of element (" << row << ", " <<
column << ") = " << res << "\n";
    mutex.unlock();
}

/* Calculates algebraical complement for the given vector of matrix
elements. */

```

```

void getSeriesAlgebraicalComplement(Matrix& matrix, std::vector <
std::pair<int, int>>& elements) {
    for (size_t i = 0; i < elements.size(); ++i) {
        getAlgebraicalComplement(matrix, elements[i].first,
elements[i].second);
    }
}

/* Consequently provides coordinates of matrix elements.
* Moves from the left to the right, then changes row. */
struct CoordinateIterator {
public:
    CoordinateIterator(int size) : size_(size) {};
    std::pair<int, int> getCoordinate() {
        std::pair<int, int> res = std::make_pair(first_, second_);
        if (second_ >= size_) {
            if (first_ < size_ - 1) {
                ++first_;
                second_ = 1;
                return std::make_pair(first_, 0);
            }
            else {
                return std::make_pair(-1, -1);
            }
        }
        ++second_;
        return res;
    }

private:
    int size_;
    int first_ = 0, second_ = 0;
};

/* Checks if the string might be converted to positive integer. */
bool isNumber(const std::string& s) {
    return !s.empty() && std::all_of(s.begin(), s.end(), ::isdigit);
}

/* Checks validity of provided arguments. */
bool checkInputValidity(int argc, char* argv[]) {
    if (argc != 3) {
        std::cout << "Invalid number of arguments! Got: " << argc - 1 <<
".\n";
        return false;
    }
    if (!isNumber(argv[1]) || std::stoi(argv[1]) < 1) {
        std::cout << "Thread number should be integer greater than 0! Got: "
<< argv[1] << ".\n";
        return false;
    }
    if (!isNumber(argv[2]) || std::stoi(argv[2]) < 2) {
        std::cout << "Matrix size should be 2 or greater! Got: " << argv[2]
<< ".\n";
        return false;
    }
}

```

```

    }
    return true;
}

int main(int argc, char* argv[])
{
    // VALIDITY CHECK SECTION
    if (!checkInputValidity(argc, argv)) {
        return 0;
    }
    int thread_number = std::stoi(argv[1]);
    int matrix_size = std::stoi(argv[2]);

    // MATRIX GENERATION SECTION
    int* arr = new int[matrix_size * matrix_size];
    for (int i = 0; i < matrix_size * matrix_size; ++i) {
        arr[i] = rand() % 200 - 100;
    }
    Matrix matrix = Matrix(matrix_size, arr);
    delete[] arr;

    std::cout << "Generated matrix is:\n";
    matrix.printMatrix();

    // DISTRIBUTION OF ELEMENTS BETWEEN THREADS
    CoordinateIterator iter(matrix_size);
    std::pair<int, int> pair;
    std::vector<std::vector<std::pair<int, int>>> pairs(thread_number);
    int cur_thread = 0;
    while ((pair = iter.getCoordinate()).first != -1) {
        pairs[cur_thread].push_back(pair);
        cur_thread = (cur_thread + 1) % thread_number;
    }

    // THREADS EXECUTION
    std::thread* threades = new std::thread[thread_number];
    for (int i = 0; i < thread_number; ++i) {
        pair = iter.getCoordinate();
        threades[i] = std::thread(getSeriesAlgebraicalComplement,
std::ref(matrix), std::ref(pairs[i]));
    }

    for (int i = 0; i < thread_number; ++i) {
        threades[i].join();
    }
}

```