

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

Факультет компьютерных наук
Образовательная программа бакалавриата «Программная инженерия»

ДОМАШНЕЕ ЗАДАНИЕ №4

Нахождение алгебраических дополнений элементов матрицы

Вариант 3

Пояснительная записка

Работу выполнил
студент группы БПИ192
Ахметьянов Арслан Рашидович
29.11.2020

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 $[-100; 100]$.

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: 2 4

Пример ожидаемого вывода:

```
Input 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 (0, 2) = -60287
Algebraical complement of element (0, 1) = -38890
Algebraical complement of element (0, 3) = 104767
Algebraical complement of element (1, 1) = 154512
Algebraical complement of element (1, 0) = 30636
Algebraical complement of element (1, 2) = 203130
Algebraical complement of element (1, 3) = 58830
Algebraical complement of element (2, 0) = -26392
Algebraical complement of element (2, 1) = -180706
Algebraical complement of element (2, 2) = -221189
Algebraical complement of element (2, 3) = 22273
Algebraical complement of element (3, 0) = 65472
Algebraical complement of element (3, 2) = 203115
Algebraical complement of element (3, 3) = -39003
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:

Input matrix is:

-59 -83

-66 -100

Algebraical complement of element (0, 0) = -100

Algebraical complement of element (0, 1) = 66

Algebraical complement of element (1, 0) = 83

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

4. Ввод: 18 9

Пример ожидаемого вывода:

Input matrix is:

-93 -95 -90 -86 -90 -84 -97 -84 -100

-99 -90 -90 -92 -86 -84 -85 -97 -92

-84 -87 -94 -99 -91 -100 -97 -94 -87

-84 -85 -95 -93 -100 -85 -88 -84 -95

-86 -100 -99 -100 -93 -89 -92 -95 -90

-94 -89 -90 -84 -100 -87 -98 -95 -85

-89 -93 -98 -100 -94 -87 -92 -91 -95

-94 -84 -84 -87 -100 -97 -96 -92 -95

-88 -86 -91 -84 -88 -92 -85 -88 -89

Algebraical complement of element (1, 2) = 878053347

Algebraical complement of element (0, 2) = 651179064

Algebraical complement of element (0, 4) = -1567002594

Algebraical complement of element (0, 1) = -1137032244

Algebraical complement of element (0, 6) = 1735357854

Algebraical complement of element (0, 8) = -540177838

Algebraical complement of element (1, 0) = -1367777582

Algebraical complement of element (0, 7) = -111473566

Algebraical complement of element (0, 3) = 1871337070

Algebraical complement of element (1, 7) = 23709913

Algebraical complement of element (1, 1) = 1439006191

Algebraical complement of element (1, 6) = 1169028951

Algebraical complement of element (1, 8) = 1424643410

Algebraical complement of element (0, 0) = 632368900

Algebraical complement of element (1, 3) = 1086320614

Algebraical complement of element (1, 4) = -1949447472

Algebraical complement of element (0, 5) = -1603389384

Algebraical complement of element (1, 5) = 1454701192

Algebraical complement of element (2, 0) = -1721625302

Algebraical complement of element (2, 2) = 1498727035

Algebraical complement of element (2, 1) = 1040335395
 Algebraical complement of element (2, 3) = 1265140832
 Algebraical complement of element (2, 6) = -1828804761
 Algebraical complement of element (2, 4) = 1952330904
 Algebraical complement of element (2, 5) = -2101088294
 Algebraical complement of element (2, 7) = -1002173101
 Algebraical complement of element (3, 1) = -139263031
 Algebraical complement of element (3, 4) = -1815139280
 Algebraical complement of element (3, 3) = -1458136188
 Algebraical complement of element (2, 8) = 756550162
 Algebraical complement of element (3, 2) = -2061810005
 Algebraical complement of element (3, 0) = -1692847342
 Algebraical complement of element (3, 5) = 105948156
 Algebraical complement of element (3, 7) = -194052679
 Algebraical complement of element (3, 6) = 2017467845
 Algebraical complement of element (3, 8) = 658543136
 Algebraical complement of element (4, 0) = 63092350
 Algebraical complement of element (4, 3) = -1745785296
 Algebraical complement of element (4, 2) = -904677489
 Algebraical complement of element (4, 1) = 912240015
 Algebraical complement of element (4, 4) = -31012026
 Algebraical complement of element (4, 5) = -927023934
 Algebraical complement of element (4, 6) = -258662133
 Algebraical complement of element (4, 7) = 204833903
 Algebraical complement of element (5, 2) = 615693379
 Algebraical complement of element (4, 8) = -1632686326
 Algebraical complement of element (5, 1) = 34443059
 Algebraical complement of element (5, 0) = 1476440160
 Algebraical complement of element (5, 3) = 76668622
 Algebraical complement of element (5, 5) = -150001826
 Algebraical complement of element (5, 4) = 541587908
 Algebraical complement of element (5, 6) = 502279717
 Algebraical complement of element (5, 7) = -1214441933
 Algebraical complement of element (5, 8) = -1854376996
 Algebraical complement of element (6, 0) = -1829513480
 Algebraical complement of element (6, 2) = 1590673003
 Algebraical complement of element (6, 5) = -1720237700
 Algebraical complement of element (6, 1) = -2074599209
 Algebraical complement of element (6, 3) = -1144290648
 Algebraical complement of element (6, 4) = 149733110
 Algebraical complement of element (6, 6) = 205765609
 Algebraical complement of element (6, 7) = 899277075
 Algebraical complement of element (7, 0) = -372691024
 Algebraical complement of element (6, 8) = 144081688

Algebraical complement of element (7, 2) = 1512747032
 Algebraical complement of element (7, 1) = -1957883618
 Algebraical complement of element (7, 4) = 1863717428
 Algebraical complement of element (7, 3) = -1406205198
 Algebraical complement of element (7, 5) = -2121802158
 Algebraical complement of element (7, 6) = -1649368046
 Algebraical complement of element (8, 0) = 244241792
 Algebraical complement of element (7, 7) = -954929590
 Algebraical complement of element (7, 8) = 953679690
 Algebraical complement of element (8, 1) = 1739858768
 Algebraical complement of element (8, 3) = 1219231768
 Algebraical complement of element (8, 2) = 695145064
 Algebraical complement of element (8, 4) = 783746522
 Algebraical complement of element (8, 6) = -1815192068
 Algebraical complement of element (8, 5) = -1603896102
 Algebraical complement of element (8, 7) = -1512728152
 Algebraical complement of element (8, 8) = 419795946

5. Ввод: 5 3

Пример ожидаемого вывода:

Generated matrix is:

-59 -33 34

0 69 24

-22 58 62

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

Список источников

1. <http://ccfit.nsu.ru/arom/data/openmp.pdf>
2. <https://docs.microsoft.com/ru-ru/cpp/parallel/openmp/reference/openmp-directives?view=msvc-160>
3. <https://docplayer.ru/48706922-Lekciya-5-paradigmy-parallelnogo-programmirovaniya.html>
4. <http://softcraft.ru/edu/comparch/>
5. <http://softcraft.ru/edu/comparch/tasks/t04/>
6. 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 <string>
#include <algorithm>
#include <omp.h>

/* 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_;
};

/* 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();
#pragma omp critical
    std::cout << "Algebraical complement of element (" << row << ", " <<
column << ") = " << res << "\n";
}

/* 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;
}

Matrix* generateMatrix(int matrix_size) {
    int* arr = new int[matrix_size * matrix_size];
    for (int i = 0; i < matrix_size * matrix_size; ++i) {
        arr[i] = rand() % 200 - 100;
    }
    return new Matrix(matrix_size, arr);
}

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
    Matrix* matrix = generateMatrix(matrix_size);

    std::cout << "Input matrix is:\n";
    matrix->printMatrix();

    omp_set_num_threads(thread_number);

    CoordinateIterator iter(matrix_size);
    std::pair<int, int> pair;

#pragma omp parallel
    while ((pair = iter.getCoordinate()).first != -1) {
        getAlgebraicalComplement(*matrix, pair.first, pair.second);
    }
}

```