

# Отчёт по расчётной работе по дисциплине ПиОИВИС

Новик Полина Александровна

12 декабря 2024 г.

## Тема: Графы

### Цель

Определить, является ли вводимый граф – графом Паппа.

### Задача

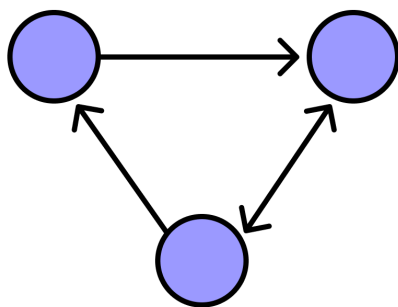
Создать программу, которая будет проверять по матрице смежности, является ли этот граф графом Паппа.

### Вариант

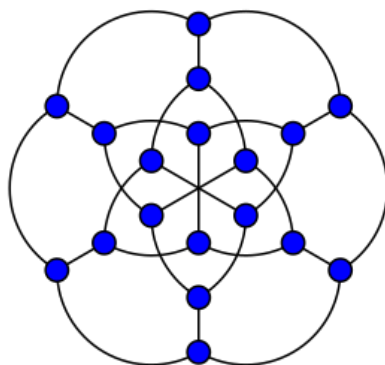
1.20(мс)

### Список ключевых понятий

- Граф — совокупность двух множеств — множества самих объектов, называемого множеством вершин, и множества их парных связей, называемого множеством рёбер. Элемент множества рёбер есть пара элементов множества вершин.



- Граф Паппа - двудольный 3-регулярный неориентированный граф с 18 вершинами и 27 рёбрами, являющийся графом Леви конфигурации Паппа.





- graph3.txt

0	1	1	1	0	0
1	0	1	0	1	0
1	1	0	0	0	1
1	0	0	0	1	1
0	1	0	1	0	1
0	0	1	1	1	0

- graph4.txt

0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1

- graph5.txt

0	0	0	0	0	0	0	0	1	0	0	0	1	1	0
0	0	0	0	0	0	0	1	0	1	0	0	0	0	1
0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1	0	0	0	0
0	1	0	0	0	1	0	0	0	1	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	1	0	0	0
0	1	0	1	0	0	0	0	0	0	0	0	1	0	0
0	0	1	0	1	0	1	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	1	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	1

## Алгоритм

1. Выбрать файл в папке и открыть его.
2. Сосчитать количество элементов и строк в файле.
3. Сравнить количество с квадратом строк матрицы.
4. Создать матрицу NxN, где N – количество строк и считывать данные с файла в эту матрицу.
5. Проверить что количество строк равно 18.
6. Проверить чтобы элемент  $a_{ij} = a_{ji}$ .
7. Проверить на то, чтобы каждая вершина имела ровно 3 связи.
8. Вывести на экран, является или нет этот граф графом Паппа.

## Код

```
#include <iostream>
#include <fstream>

using namespace std;

int main() {

    // Ввод имени файла
    string name;
    int number;
    cout << "Введите номер файла: "; cin >> number;

    switch (number) {
        case 1: name = "graph.txt"; break;
        case 2: name = "graph2.txt"; break;
        case 3: name = "graph3.txt"; break;
        case 4: name = "graph4.txt"; break;
        case 5: name = "graph5.txt"; break;
        default:
            cout << "Неправильный номер файла." << endl;
            return 0;
    }

    ifstream fin(name);
    if (!fin.is_open()) {
        cout << "Не удалось открыть файл" << endl;
        return 0;
    }

    const int MAX_size = 100; // Максимальный размер
    int matrix[MAX_size][MAX_size] = { 0 }; // Матрица смежности
    int row = 0, col = 0;

    char chm;
    int value, colInRow = 0;
    bool hasError = false; // Флаг ошибок

    // Флаги для каждой ошибки
    bool sizeError = false, matrixError = false, rowCountError = false,
        connectionError = false;

    while (fin.get(chm)) { //считывает один символ и переводит в int(число)
        if (chm == '0' || chm == '1') {
            value = chm - '0';
            matrix[row][colInRow++] = value;
        } else if (chm == '\n') {
            if (row == 0) {
                col = colInRow; // Запоминаем количество столбцов после первой строки
            } else if (colInRow != col) {
                if (!sizeError) {
                    cout << "Некорректный размер матрицы" << endl;
                    sizeError = true;
                }
            }
            row++;
            colInRow = 0;
        }
    }

    if (colInRow > 0) { // Для последней строки без перевода строки
```

```

        if (row == 0) {
            col = colInRow;
        } else if (colInRow != col) {
            if (!sizeError) {
                cout << "Некорректный размер матрицы" << endl;
                sizeError = true;
            }
        }
        row++;
    }
    fin.close();

    // Вывод размера матрицы
    cout << "Размер матрицы: " << row << "x" << col << endl;

    // Вывод матрицы смежности
    cout << "Матрица смежности:" << endl;
    for (int i = 0; i < row; i++) {
        for (int j = 0; j < col; j++) {
            cout << matrix[i][j] << " ";
        }
        cout << endl;
    }

    // Проверка значений матрицы (0 или 1)
    for (int i = 0; i < row; i++) {
        for (int j = 0; j < col; j++) {
            if (matrix[i][j] != 0 && matrix[i][j] != 1) {
                if (!matrixError) {
                    cout << "Матрица должна состоять из 0 и 1!" << endl;
                    matrixError = true;
                }
            }
        }
    }

    // Проверка количества строк (вершин)
    if (row != 18) {
        if (!rowCountError) {
            cout << "Матрица должна содержать 18 строк (вершин)!" << endl;
            rowCountError = true;
        }
    }

    // Проверка кубичности (каждая вершина имеет ровно 3 связи)
    for (int i = 0; i < row; i++) {
        int connections = 0;
        for (int j = 0; j < row; j++) {
            if (matrix[i][j] == 1) {
                connections++;
            }
        }
        if (connections != 3) {
            if (!connectionError) {
                cout << "Каждая вершина должна иметь ровно 3 связи!" << endl;
                connectionError = true;
            }
        }
    }

    // Проверка количества рёбер
    int edgeCount = 0;
    for (int i = 0; i < row; i++) {

```

```

        for (int j = i + 1; j < row; j++) {
            if (matrix[i][j] == 1) {
                edgeCount++;
            }
        }
    }
    if (edgeCount != 27) {
        cout << "Матрица должна содержать 27 рёбер!" << endl;
        hasError = true;
    }

    // Если хотя бы одна ошибка была выведена
    if (sizeError || matrixError || rowCountError || connectionError || hasError) {
        cout << "Это не граф Паппа!" << endl;
    } else {
        cout << "Это граф Паппа!" << endl;
    }

    return 0;
}

```

## Вывод

В результате были изучены:

- основы теории графов;
- различные виды графов;
- способы представления графов;
- алгоритмы для работы с графами.