|  |
| --- |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **"МИРЭА – Российский технологический университет"**  **РТУ МИРЭА** |

Институт комплексной безопасности и специального приборостроения

Кафедра КБ-14 «Цифровые технологии обработки данных»

**ОТЧЕТ   
о выполнении лабораторной работы №3**

**«Алгоритмы на графах»**

**по дисциплине   
«Алгоритмы и структуры данных»**

**Вариант № 55**

Выполнил: студент 2 курса

группы \_\_\_\_\_\_\_\_\_\_\_

шифр \_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_

*(фио студента)*

Проверил: *\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

Москва 2021 г.

**Вариант № 55.**

**Алгоритм:** Определить, есть ли какой-либо путь, проходящий через ВСЕ   
вершины орграфа, причем через вершину можно проходить только   
один раз, а начальная и конечная вершины должны совпадать, и   
вывести его на экран

**Способ представления графа:** Матрица смежности.

**Теория о Графах.**

Граф — это математический объект, который состоит из точек и линий, которые их соединяют. Точки называют вершинами графа, а линии — ребрами. Граф, ребра которого имеют направления, называется ориентированным, если же ребра графа не имеют направления, то такой граф называется неориентированным.



Рисунок 1. – Пример ориентированного графа.

Цикл графа – некоторое число вершин, соединенных замкнутой цепью. Так, например, на графе, изображенном на рисунке 1, циклами будут являться следующие соединенные вершины: A->B->C->D->A, A->B->C->E->F->A, A->B->D->A, A->G->D->A, B->C->E->B и B->C->E->F->B.

Матрица смежности — это вид представления графа в виде матрицы, когда пересечение столбцов и строк задаёт дуги. Используя матрицу смежности, можно задать вес дуг и ориентацию. Каждая строка и столбец матрицы соответствуют вершинам, номер строки соответствует вершине, из которой выходит дуга, а номер столбца - в какую входит дуга. Пример матрицы смежности графа, изображенного на рисунке 1, представлен на рисунке 2.



Рисунок 2. – Пример матрицы смежности.

Для поиска в орграфе маршрута проходящего через все вершины поспользуемся поиском в глубину. Если в процессе у нас получится путь длинной равной количеству вершин + 1 и точка начала будет концом, то такой путь найден. В противном случае такого пути не существует.

**Листинг программы.**

def load\_matrix(filename):

    with open(filename) as f:

        matrix = []

        for line in f.readlines():

            matrix.append(list(map(int, line.split())))

        return matrix

def DFS(matrix):

    path\_stack = []

    def \_dfs(v1: int):

        path\_stack.append(v1)

        for v2, e in enumerate(matrix[v1]):

            if e:

                if v2 in path\_stack and v2 == path\_stack[0] and len(path\_stack) == len(matrix):

                    path\_stack.append(v2)

                    return True

                elif v2 not in path\_stack:

                    if \_dfs(v2):

                        return True

        path\_stack.remove(v1)

    for i in range(len(matrix)):

        if \_dfs(i):

            return path\_stack

    return []

matrix = load\_matrix('graph\_task55.txt')

path = DFS(matrix)

if path:

    print("Путь проходящий через все вершины графа:")

    print("->".join(map(str, path)))

else:

    print("В данном графе не существует цикла проходящего через все вершины графа!")

**Скриншот работы программы:**

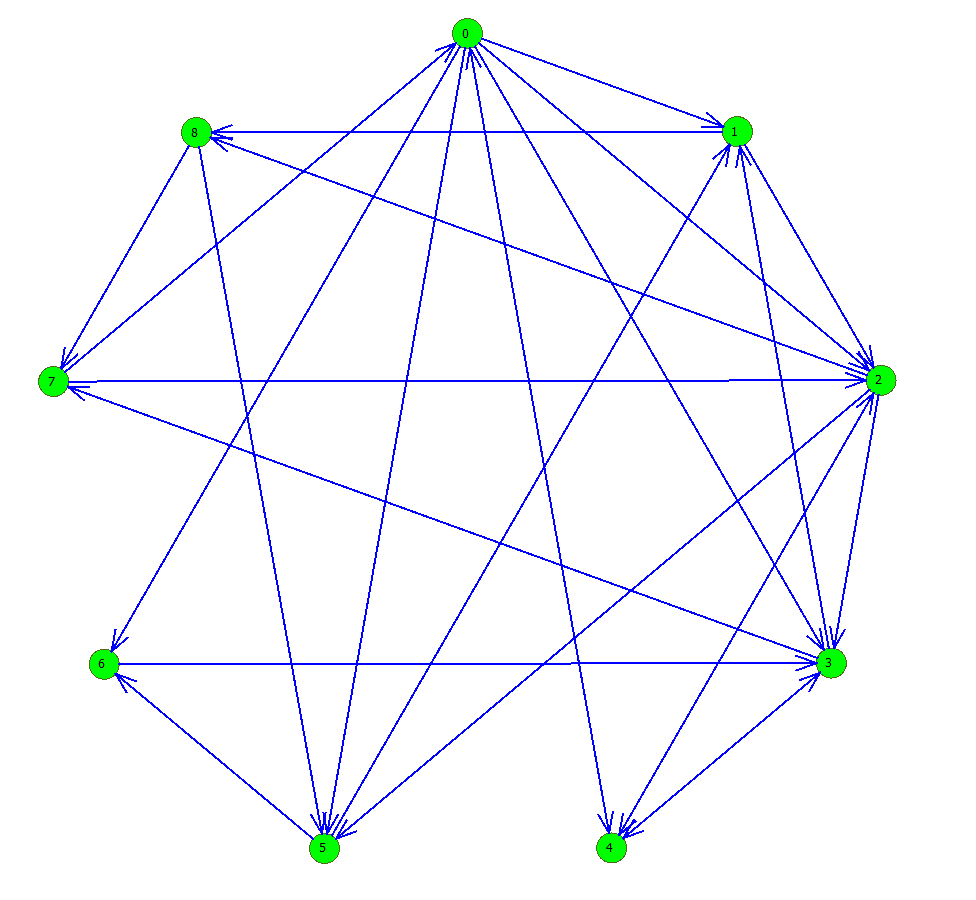


Рисунок 3. – Пример ненагруженного орграфа графа с циклом.

****

Рисунок 4. – Пример работы программы.

**Выводы.**

В результате выполнения данной работы была изучена структура граф, ее свойства и операции над ней. Так же был реализован алгоритм поиска в глубину на для поиска маршрута проходящего через все вершины.

**Литература:**

1. Алгоритмы: построение и анализ. Кормен Т.Х., Лейзерсон Ч.И.

2. https://en.wikipedia.org/wiki/Cycle\_(graph\_theory)

3. https://en.wikipedia.org/wiki/Depth-first\_search