|  |
| --- |
| 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 № 33**

Выполнил: студент 2 курса

группы \_\_\_\_\_\_\_\_\_\_\_

шифр \_\_\_\_\_\_\_\_\_\_\_\_

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

*(фио студента)*

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

Москва 2021 г.

**Вариант № 33.**

**Алгоритм:** определить наличие всех циклов методом обхода в глубину на  
орграфе. Вывести все циклы (варианты обхода, образующие  
циклы). Подсчитать их общее количество.

**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 всех циклов в ориентированном графе воспользуемся поиском в глубину. Применим его к каждой вершине на графе и уберем из списка пути, конец которых не совпадает с началом.

**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(adj\_matrix: list[list[int]]) -> list[list[int]]:

    def \_dfs(node: int, known\_nodes: set[int]):

        paths = []

        for adj\_node, adj in enumerate(adj\_matrix[node]):

            if adj <= 0:

                continue

            if adj\_node in known\_nodes:

                paths.append([node, adj\_node])

                continue

            a = \_dfs(adj\_node, known\_nodes | set([adj\_node]))

            paths.extend([[node, \*path] for path in a])

        return paths if paths else [[node]]

    known\_nodes = set()

    paths = []

    for node in range(len(adj\_matrix)):

        if node not in known\_nodes:

            known\_nodes.add(node)

            paths.extend(\_dfs(node, set([node])))

    return paths

def find\_cycles(paths: list[list[int]]):

    return [path for path in paths if path[0] == path[-1]]

matrix = load\_matrix('graph\_task33.txt')

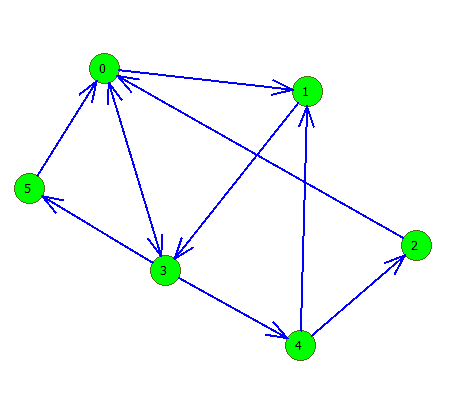
paths = DFS(matrix)

cycles = find\_cycles(paths)

print("Количество циклов:", len(cycles))

print("Циклы:")

print("\n".join(["->".join(map(str, cycle)) for cycle in cycles]))

**Скриншот работы программы:**

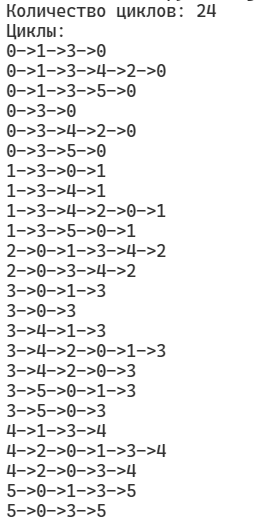
Рисунок 3. – Пример ориентированного ненагруженного графа.

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 всех циклов ориентированном графе.

**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