МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Український державний університет імені Михайла Драгоманова

Факультет математики, інформатики та фізики

Кафедра інформаційних технологій та програмування

**Звіт**

з лабораторної роботи №9

«Класичні алгоритми опрацювання масивів»

з дисципліни «Програмування»

Виконав:

студент ІІІ курсу групи 31І

Дзюба В.І.

Перевірила:

викладач Устименко О.Б.

Оцінка\_\_\_\_\_\_\_\_\_\_\_

Дата \_\_\_\_\_\_\_\_\_\_\_\_\_

**Київ – 2024**

**Зміст**

[Мета роботи 3](#_Toc168149831)

[Постановка задачі 4](#_Toc168149832)

[Основна частина 5](#_Toc168149833)

[2.1 Опис вхідних та вихідних даних 5](#_Toc168149834)

[2.2 Блок-схема 6](#_Toc168149835)

[Висновки 7](#_Toc168149836)

[Додатки 8](#_Toc168149837)

# Мета роботи

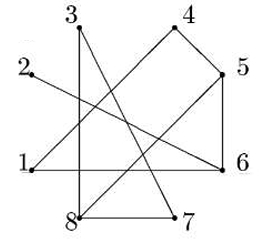
Метою цієї лабораторної роботи є набуття навичок реалізації та застосування алгоритмів обходу графів на прикладі алгоритму пошуку в глибину (DFS), зокрема, навчитися створювати та представляти графи, реалізовувати алгоритм DFS, а також перевіряти правильність його роботи на заданому графі.

# Постановка задачі

Варіант №12

**Завдання**

Написати програму для обходу графа за алгоритмом пошуку в глибину



# Основна частина

## 2.1 Опис вхідних та вихідних даних

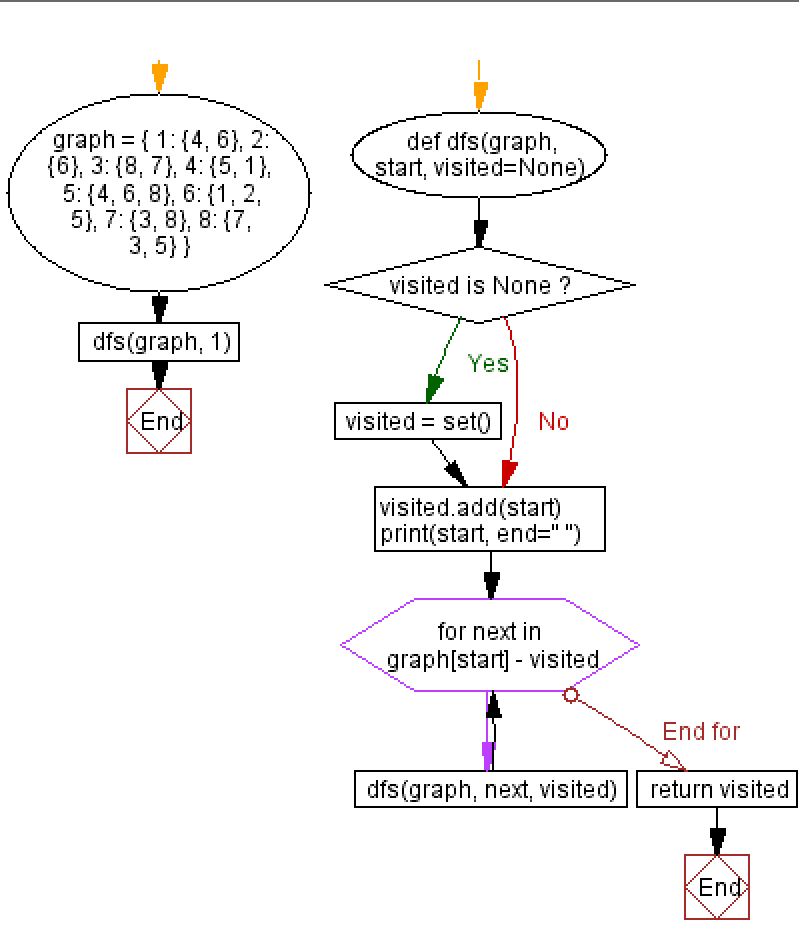
**Вхідні дані:**

1. **graph**: Незважений і не орієнтований граф, представлений у вигляді словника, де ключі - це вершини графа, а значення - множини суміжних вершин.
2. **start**: Початкова вершина, з якої починається обхід графа.

**Вихідні дані:**

1. **visited**: Порядок обходу вершин графа, виведений на екран у процесі виконання алгоритму DFS.
2. алгоритму DFS.

## 2.2 Блок-схема



# Висновки

Виконана лабораторна робота дозволила закріпити знання з алгоритмів обходу графів, зокрема алгоритму пошуку в глибину, шляхом створення графа, реалізації алгоритму DFS, перевірки правильності його роботи, та покращення навичок програмування і роботи з графовими структурами даних на Python.

# Додатки

# Варіант 12

#

# Написати програму для обходу графа за алгоритмом пошуку в глибину

#

# Дзюба Владислав

def dfs(graph, start, visited=None):

    if visited is None:

        visited = set()

    visited.add(start)

    print(start, end=" ")

    for next in graph[start] - visited:

        dfs(graph, next, visited)

    return visited

graph = {

    1: {4, 6},

    2: {6},

    3: {8, 7},

    4: {5, 1},

    5: {4, 6, 8},

    6: {1, 2, 5},

    7: {3, 8},

    8: {7, 3, 5}

}

dfs(graph, 1)

**Результат:**

****