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 №10

«Графи, динамічне програмування, жадібні алгоритми»

з дисципліни «Програмування»

Виконав:

студент ІІІ курсу групи 31І

Буртенко К.С.

Перевірила:

викладач Устименко О.Б.

Оцінка\_\_\_\_\_\_\_\_\_\_\_

Дата \_\_\_\_\_\_\_\_\_\_\_\_\_

**Київ – 2024**

**Зміст**

[Мета роботи 3](#_Toc162344475)

[Постановка задачі 4](#_Toc162344476)

[Основна частина 5](#_Toc162344477)

[2.1 Опис вхідних та вихідних даних 5](#_Toc162344478)

[2.2 Блок-схема 6](#_Toc162344479)

[Висновки 7](#_Toc162344480)

[Додатки 8](#_Toc162344481)

# Мета роботи

Мета лабораторної роботи полягає у вивченні та практичному застосуванні основних алгоритмів в даній темі та написанні двох програм, що демонструють ці принципи.

# Постановка задачі

Варіант №5

**Завдання**

1. Написати програму для пошуку найкоротшого шляху з кожної вершини до усіх інших вершин:

Зображення, що містить ряд

Автоматично згенерований опис

1. Дані послідовність чисел. Знайти довжину найдовшої строго зростаючої підпослідовності.

# Основна частина

## 2.1 Опис вхідних та вихідних даних

**Завдання 1**

**Вхідні дані:**

Данні про граф, його будова. Як вершини з’єднані між собою

**Вихідні дані:**

Будується найкоротший шлях від одної вершини до іншої

**Результат роботи програми:**

**Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис**

**Завдання 2.**

**Вхідні дані:**

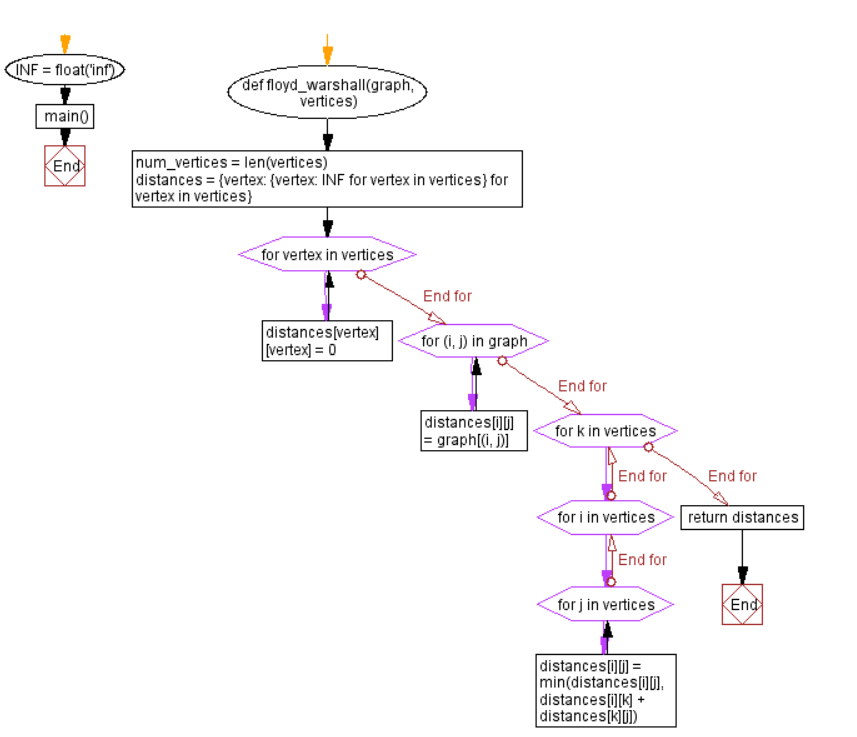
Послідовність чисел

**Вихідні дані:**

Довжина найдовшої строго зростаючої підпослідовності

## 2.2 Блок-схема

**Завдання 1**



Зображення, що містить текст, схема, знімок екрана, ряд

Автоматично згенерований опис

**Завдання 2.**

Зображення, що містить текст, схема, знімок екрана, ряд

Автоматично згенерований опис

# Висновки

Під час виконання лабораторної були використані навички з теми «Класичні алгоритми опрацювання масивів». Створено програму, яка показує сортування рядків двовимірного масиву методом злиття.

# Додатки

**Завдання 1.**

INF = float('inf')

def floyd\_warshall(graph, vertices):

num\_vertices = len(vertices)

distances = {vertex: {vertex: INF for vertex in vertices} for vertex in vertices}

for vertex in vertices:

distances[vertex][vertex] = 0

for (i, j) in graph:

distances[i][j] = graph[(i, j)]

for k in vertices:

for i in vertices:

for j in vertices:

distances[i][j] = min(distances[i][j], distances[i][k] + distances[k][j])

return distances

def print\_solution(distances):

vertices = list(distances.keys())

print("Матриця найкоротших відстаней між усіма парами вершин:")

for i in vertices:

for j in vertices:

if distances[i][j] == INF:

print("INF", end=" ")

else:

print(distances[i][j], end=" ")

print()

def main():

graph = {

('a', 'b'): 1,

('b', 'd'): 5,

('b', 'c'): 2,

('b', 'z'): 7,

('c', 'a'): 2,

('c', 'e'): 4,

('c', 'd'): 1,

('d', 'z'): 1,

('e', 'c'): 3,

('z', 'e'): 1

}

vertices = ['a', 'b', 'c', 'd', 'e', 'z']

distances = floyd\_warshall(graph, vertices)

print\_solution(distances)

main()

**Завдання 2.**

def longest\_increasing\_subsequence\_length(nums):

if not nums:

return 0

n = len(nums)

dp = [1] \* n

for i in range(1, n):

for j in range(i):

if nums[j] > nums[j - 1]:

dp[i] = max(dp[i], dp[j] + 1)

print(dp)

return max(dp)

nums = [15, 7, 24, 25, 7, 19, 55, 64, 78]

print("Довжина найдовшої строго зростаючої підпослідовності:", longest\_increasing\_subsequence\_length(nums))