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»





Київ - 2024

**Зміст**

[Мета роботи 3](#_heading=h.30j0zll)

[1 Постановка задачі 4](#_heading=h.1fob9te)

[2 Основна частина 4](#_heading=h.3znysh7)

[2.1 Опис вхідних та вихідних даних](#_heading=h.2et92p0)

2.2 Блок-схема [5](#_heading=h.tyjcwt)

[Висновки 6](#_heading=h.3dy6vkm)

[Список літератури 7](#_heading=h.1t3h5sf)

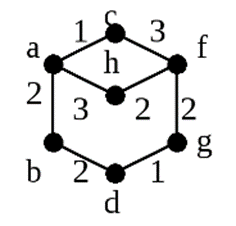
[Додатки 8](#_heading=h.4d34og8)

**Мета роботи**

Мета лабораторної роботи – використання модуля turtle у Python для вивчення основ графічного програмування.

**1 Постановка задачі**

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



1. Дано два рядки. Знайти довжину найбільшого підрядка, який є підрядком обох рядків.

**2 Основна частина**

**2.1 Опис вхідних та вихідних даних**

**1 задача**

Вхідні дані:

Опис функції dijkstra

* graph: Словник, що представляє граф.

graph = {

'a': {'h': 3, 'b': 2, 'c': 1},

'b': {'a': 2, 'd': 2},

'c': {'a': 1, 'f': 3},

'd': {'b': 2, 'g': 1},

'f': {'c': 3, 'g': 2, 'h': 2},

'g': {'f': 2, 'd': 1},

'h': {'a': 3, 'f': 2}

}

* start: Вершина, з якої починається обхід графа для знаходження найкоротших шляхів до всіх інших вершин.

Вихідні дані:

* distances: Словник, де ключами є вершини графа, а значеннями — найкоротші відстані від стартової вершини до кожної з вершин.

distances = {

'a': 0,

'h': 3,

'b': 2,

'c': 1,

'd': 4,

'f': 4,

'g': 5

}

Опис функції find\_all\_shortest\_paths

Вхідні дані:

* graph: Словник, що представляє граф. (той же формат, що й для dijkstra)

Вихідні дані:

* shortest\_paths: Словник, де ключами є вершини графа, а значеннями — словники найкоротших відстаней від відповідної вершини до всіх інших вершин (результати роботи dijkstra для кожної вершини).

shortest\_paths = {

'a': {'a': 0, 'h': 3, 'b': 2, 'c': 1, 'd': 4, 'f': 4, 'g': 5},

'b': {'a': 2, 'h': 5, 'b': 0, 'c': 3, 'd': 2, 'f': 5, 'g': 3},

'c': {'a': 1, 'h': 4, 'b': 3, 'c': 0, 'd': 4, 'f': 3, 'g': 5},

'd': {'a': 4, 'h': 7, 'b': 2, 'c': 5, 'd': 0, 'f': 6, 'g': 1},

'f': {'a': 4, 'h': 2, 'b': 5, 'c': 3, 'd': 6, 'f': 0, 'g': 2},

'g': {'a': 5, 'h': 4, 'b': 3, 'c': 5, 'd': 1, 'f': 2, 'g': 0},

'h': {'a': 3, 'h': 0, 'b': 5, 'c': 4, 'd': 7, 'f': 2, 'g': 4}

}

**2 задача**

Опис функції common

Вхідні дані:

* str1: Перший рядок для порівняння. Наприклад, "abcdef".
* str2: Другий рядок для порівняння. Наприклад, "zabcdf".

Вихідні дані:

* Ціле число, що представляє довжину найбільшого спільного підрядка між двома рядками. Наприклад, для рядків "abcdef" та "zabcdf" вихідне значення буде 4, оскільки найбільший спільний підрядок — "abcd".

**Висновки**

У цій лабораторній роботі я дослідила основні алгоритмічні підходи, зокрема, алгоритми роботи з графами, методи динамічного програмування та жадібні алгоритми. Застосувавши алгоритм Дейкстри для знаходження найкоротших шляхів у графах, а також розглянувши динамічне програмування для задачі знаходження найбільшого спільного підрядка, я зрозуміла їхню важливість та ефективність у розв'язанні складних обчислювальних задач у практичному програмуванні.

**Список літератури**

* <https://ua5.org/algorithm/1909-zhadibni-algorytmy.html>
* <https://uk.wikipedia.org/wiki/%D0%94%D0%B8%D0%BD%D0%B0%D0%BC%D1%96%D1%87%D0%BD%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F>
* <https://foxminded.ua/metod-dynamichnoho-prohramuvannia/>

**Додатки**

Лістинги програм

