МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Львівський національний університет імені Івана Франка

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ ТА ІНФОРМАТИКИ

Кафедра дискретного аналізу та інтелектуальних систем

**Звіт з лабораторної роботи №10**

**“Алгоритм дейкстри”**

Роботу виконав:

**Тимчишин Ярема Андрійович**

Студент групи Пмі-13

Перевірила:

**Корольчук Світлана Ярославівна,** завідувач Лабораторії програмування, асистент кафедри програмуванняЛьвівського національного

університету імені Івана Франка

Львів – 2022

**ЗМІСТ**

ВСТУП 3

РОЗДІЛ 1 9

***1.1. Створення допоміжної функції, яка шукатиме вершину з мінімальним значенням відстані.*** 9

***1.2. Створення допоміжної функції, яка буде друкувати створений масив відстаней.*** 9

***1.3. Функція, яка реалізує алгоритм Дейкстри*** 10

РОЗДІЛ 2 11

***2.1. Ініціалізація графу за допомогою якого перевіримо справність роботи алгоритму*** 11

***2.2. Результат*** 11

ВИСНОВКИ 12

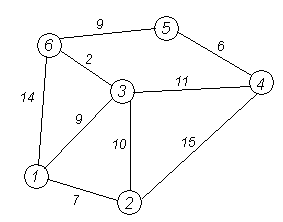
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 12

# ВСТУП

**Алгоритм Дейкстри** — алгоритм на графах, відкритий Дейкстрою. Знаходить найкоротший шлях від однієї вершини графу до всіх інших вершин. Класичний алгоритм Дейкстри працює тільки для графів без ребер від'ємної довжини.

**Приклад роботи алгоритму**

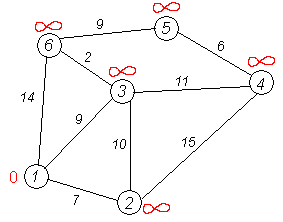
Зберігатимемо поточну мінімальну відстань до всіх вершин **V** (від даної вершини **a**) і на кожному кроці алгоритму намагатимемося зменшити цю відстань. Спочатку встановимо відстані до всіх вершин рівними нескінченості, а до вершини **а** — нулю. Розглянемо виконання алгоритму на прикладі. Хай потрібно знайти відстані від 1-ї вершини до всіх інших.



Кружечками позначені вершини, лініями — шляхи між ними («дуги»). Над дугами позначена їх «ціна» — довжина шляху. Надписом над кружечком позначена поточна найкоротша відстань до вершини.

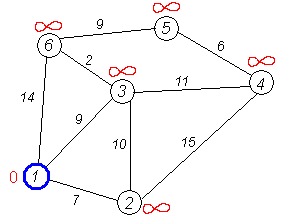
**Крок 1**

Ініціалізація. Відстань до всіх вершин графу **V** = {\displaystyle \infty }. Відстань до **а** = 0. Жодної вершини графу ще не опрацьовано.



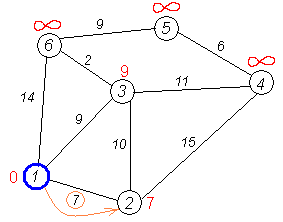
**Крок 2**

Знаходимо таку вершину (із ще не опрацьованих), поточна найкоротша відстань до якої мінімальна. В нашому випадку це вершина 1. Обходимо всіх її сусідів і, якщо шлях в сусідню вершину через 1 менший за поточний мінімальний шлях в цю сусідню вершину, то запам'ятовуємо цей новий, коротший шлях як поточний найкоротший шлях до сусіда.



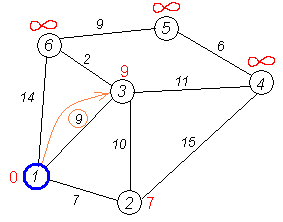
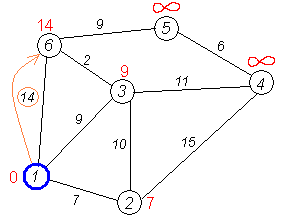
**Крок 3**

Перший по порядку сусід 1-ї вершини — 2-а вершина. Шлях до неї через 1-у вершину дорівнює найкоротшій відстані до 1-ї вершини + довжина дуги між 1-ю та 2-ю вершиною, тобто 0 + 7 = 7. Це менше поточного найкоротшого шляху до 2-ї вершини, тому найкоротший шлях до 2-ї вершини дорівнює 7.



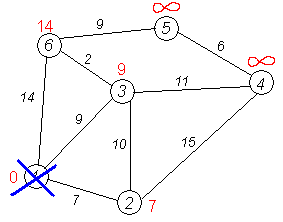
**Кроки 4, 5**

Аналогічну операцію проробляємо з двома іншими сусідами 1-ї вершини — 3-ю та 6-ю.

[](https://uk.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:Dijkstra_graph4.PNG) [](https://uk.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:Dijkstra_graph5.PNG)

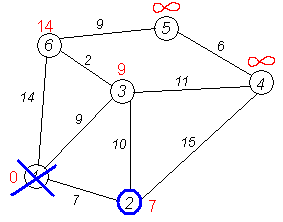
**Крок 6**

Всі сусіди вершини 1 перевірені. Поточна мінімальна відстань до вершини 1 вважається остаточною і обговоренню не підлягає. Тому викреслимо її з графу, щоб відмітити цей факт.



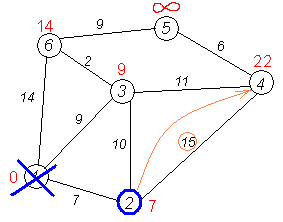
**Крок 7**

Практично відбувається повернення до кроку 2. Знову знаходимо «найближчу» необроблену (невикреслену) вершину. Це вершина 2 з поточною найкоротшою відстанню до неї = 7. І знову намагаємося зменшити відстань до всіх сусідів 2-ї вершини, намагаючись пройти в них через 2-у. Сусідами 2-ї вершини є 1, 3, 4.



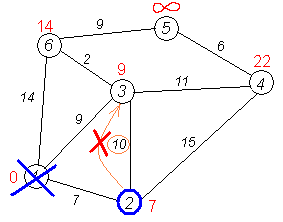
**Крок 8**

Перший (по порядку) сусід вершини № 2 — 1-ша вершина. Але вона вже оброблена (або викреслена — див. крок 6). Тому з 1-ю вершиною нічого не робимо. Інший сусід вершини 2 — вершина 4. Якщо йти в неї через 2-у, то шлях буде = найкоротша відстань до 2-ї + відстань між 2-ю і 4-ю вершинами = 7 + 15 = 22. Оскільки 22 < ∞, встановлюємо відстань до вершини № 4 рівним 22.



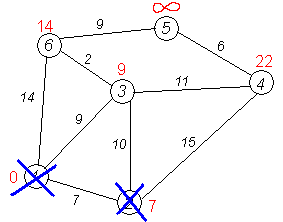
**Крок 9**

Ще один сусід вершини 2 — вершина 3. Якщо йти в неї через 2-у, то шлях буде = 7 + 10 = 17. Але 17 більше за відстань, що вже запам'ятали раніше до вершини № 3 і дорівнює 9, тому поточну відстань до 3-ї вершини не міняємо.



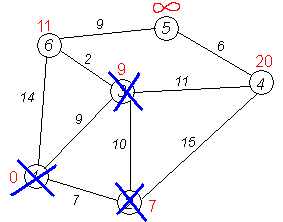
**Крок 10**

Всі сусіди вершини 2 переглянуті, заморожуємо відстань до неї і викреслюємо її з графу.



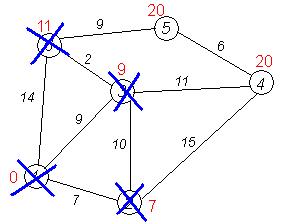
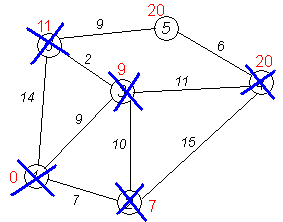
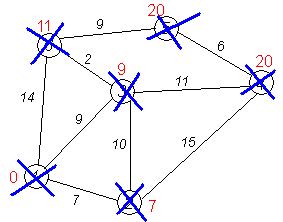
**Крок 11 – 15**

По вже «відпрацьованій» схемі повторюємо кроки 2 — 6. Тепер «найближчою» виявляється вершина № 3. Після її «обробки» отримаємо такі результати:



**Наступні кроки**

Проробляємо те саме з вершинами, що залишилися (№ по порядку: 6, 4 і 5).

[](https://uk.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:Dijkstra_graph12.PNG)[](https://uk.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:Dijkstra_graph13.PNG)[](https://uk.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:Dijkstra_graph14.PNG)

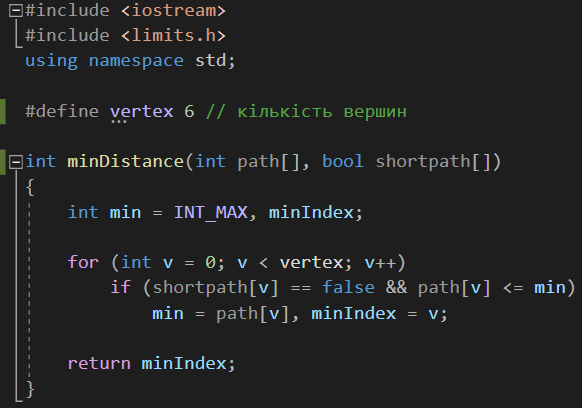
**Завершення виконання алгоритму**

Алгоритм закінчує роботу, коли викреслені всі вершини. Результат його роботи видно на останньому малюнку: найкоротший шлях від 1-ї вершини до 2-ї становить 7, до 3-ї — 9, до 4-ї — 20, до 5-ї — 20, до 6-ї — 11 умовних одиниць.

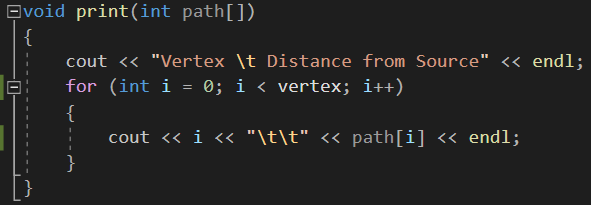
# РОЗДІЛ 1

**Написання головної програми**

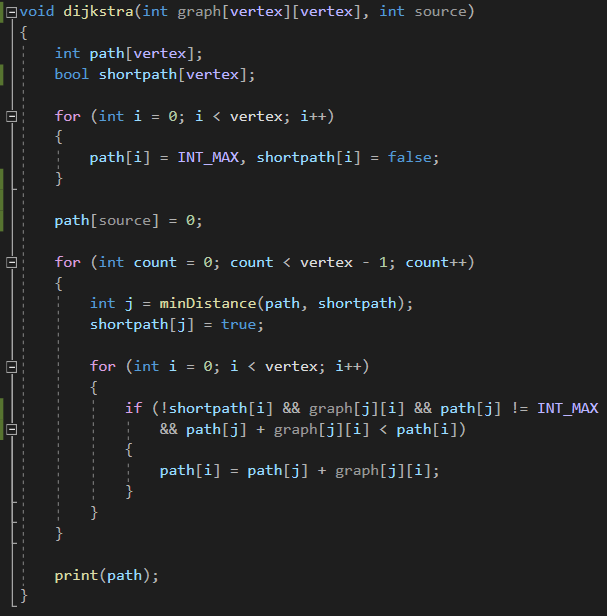
## *1.1. Створення допоміжної функції, яка шукатиме вершину з мінімальним значенням відстані.*



## *1.2. Створення допоміжної функції, яка буде друкувати створений масив відстаней.*



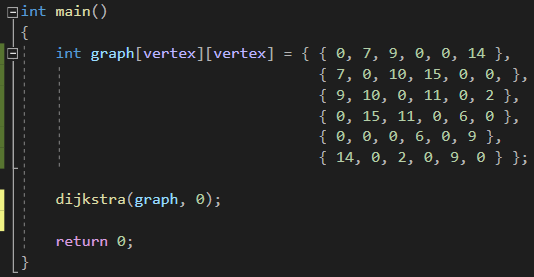
## *1.3. Функція, яка реалізує алгоритм Дейкстри*



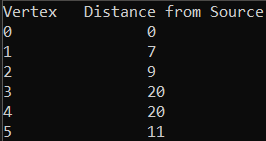
# РОЗДІЛ 2

**Перевірка справності написаної програми**

## *2.1. Ініціалізація графу за допомогою якого перевіримо справність роботи алгоритму*



## *2.2. Результат*



# ВИСНОВКИ

У результаті виконання роботи:

1. Розроблено програму для реалізації алгоритму Дейкстри.
2. Перевірено правильність виконання написаної програми.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Режим доступу: [Алгоритм Дейкстри — Вікіпедія (wikipedia.org)](https://uk.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%94%D0%B5%D0%B9%D0%BA%D1%81%D1%82%D1%80%D0%B8)