**Міністерство освіти і науки України**

**Львівський національний університет імені Івана Франка**

Факультет прикладної математики та інформатики

*Звіт*

*Лабораторна робота №10*

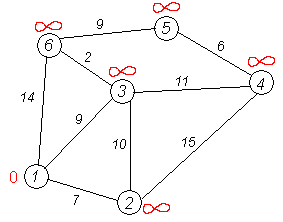
«Алгоритм Дейкстри»  
**з дисципліни «Алгоритми та структури даних»**

Виконав студент групи ПМІ-11  
 Яцуляк Андрій

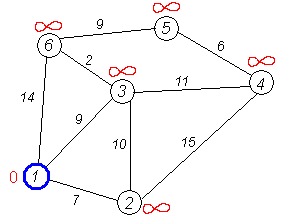
**Алгоритм Дейкстри**

**Алгоритм Дейкстри** — [алгоритм](https://uk.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC) на [графах](https://uk.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)), відкритий [Дейкстрою](https://uk.wikipedia.org/wiki/%D0%94%D0%B5%D0%B9%D0%BA%D1%81%D1%82%D1%80%D0%B0_%D0%95%D0%B4%D1%81%D0%B3%D0%B5%D1%80" \o "Дейкстра Едсгер). Знаходить [найкоротший шлях](https://uk.wikipedia.org/wiki/%D0%9D%D0%B0%D0%B9%D0%BA%D0%BE%D1%80%D0%BE%D1%82%D1%88%D0%B8%D0%B9_%D1%88%D0%BB%D1%8F%D1%85) від однієї вершини графу до всіх інших вершин. Класичний алгоритм Дейкстри працює тільки для графів без ребер від'ємної довжини.

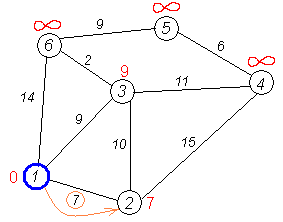
**Крок 1**

Ініціалізація. Відстань до всіх вершин [графу](https://uk.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) **V** = {\displaystyle \infty }. Відстань до **а** = 0. Жодної вершини графу ще не опрацьовано.  
[](https://uk.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:Dijkstra_graph1.PNG)

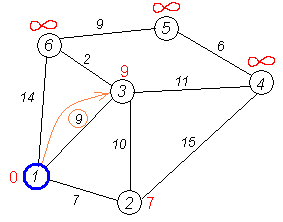
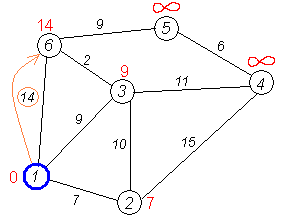
**Крок 2**

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.  
[](https://uk.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:Dijkstra_graph2.PNG)

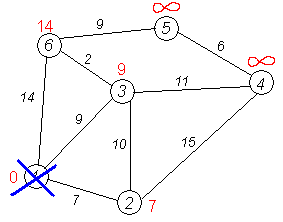
**Крок 3**

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.  
[](https://uk.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:Dijkstra_graph3.PNG)

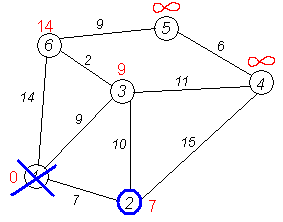
**Кроки 4, 5**

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 (те, що це дійсно так, вперше довів [Дейкстра](https://uk.wikipedia.org/wiki/%D0%94%D0%B5%D0%B9%D0%BA%D1%81%D1%82%D1%80%D0%B0_%D0%95%D0%B4%D1%81%D0%B3%D0%B5%D1%80" \o "Дейкстра Едсгер)). Тому викреслимо її з [графу](https://uk.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)), щоб відмітити цей факт. [](https://uk.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:Dijkstra_graph6.PNG)

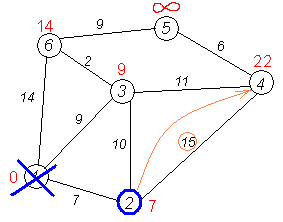
**Крок 7**

Практично відбувається повернення до кроку 2. Знову знаходимо «найближчу» необроблену (невикреслену) вершину. Це вершина 2 з поточною найкоротшою відстанню до неї = 7. [](https://uk.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:Dijkstra_graph7.PNG) 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.

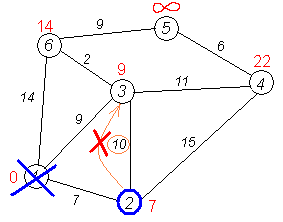
**Крок 8**

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.

**Крок 8 (з іншими вхідними даними)**

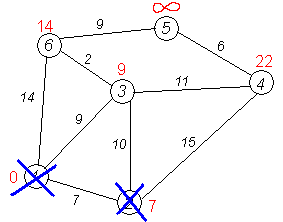
Інший сусід вершини 2 — вершина 4. Якщо йти в неї через 2-у, то шлях буде = найкоротша відстань до 2-ї + відстань між 2-ю і 4-ю вершинами = 7 + 15 = 22. Оскільки 22 < ∞, встановлюємо відстань до вершини № 4 рівним 22. [](https://uk.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:Dijkstra_graph8.PNG)

**Крок 9**

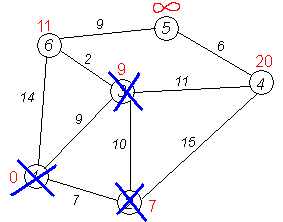
Ще один сусід вершини 2 — вершина 3. Якщо йти в неї через 2-у, то шлях буде = 7 + 10 = 17. Але 17 більше за відстань, що вже запам'ятали раніше до вершини № 3 і дорівнює 9, тому поточну відстань до 3-ї вершини не міняємо. [](https://uk.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:Dijkstra_graph9.PNG)

**Крок 10**

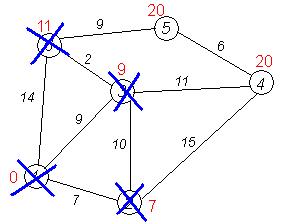
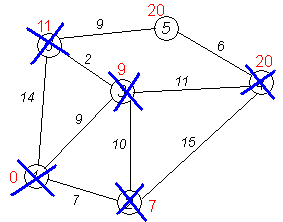
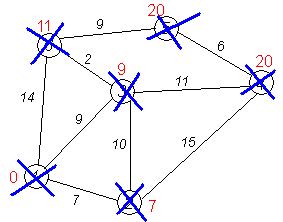
Всі сусіди вершини 2 переглянуті, заморожуємо відстань до неї і викреслюємо її

з [графу](https://uk.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)). [](https://uk.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:Dijkstra_graph10.PNG)

**Кроки 11 — 15**

По вже «відпрацьованій» схемі повторюємо кроки 2 — 6. Тепер «найближчою» виявляється вершина № 3. Після її «обробки» отримаємо такі результати:  
[](https://uk.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:Dijkstra_graph11.PNG)

**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.