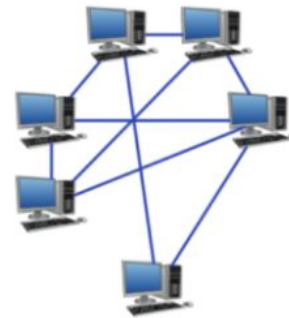
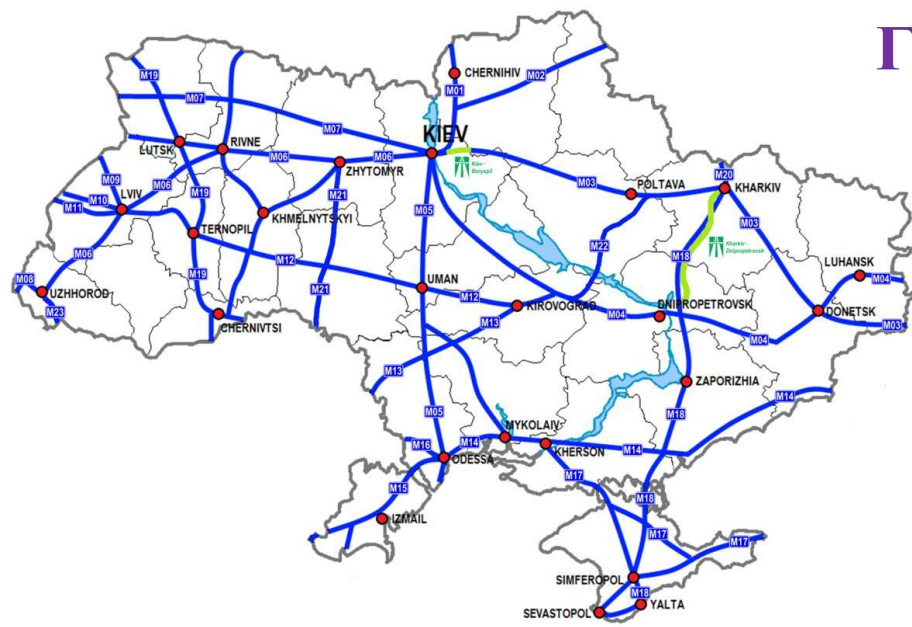
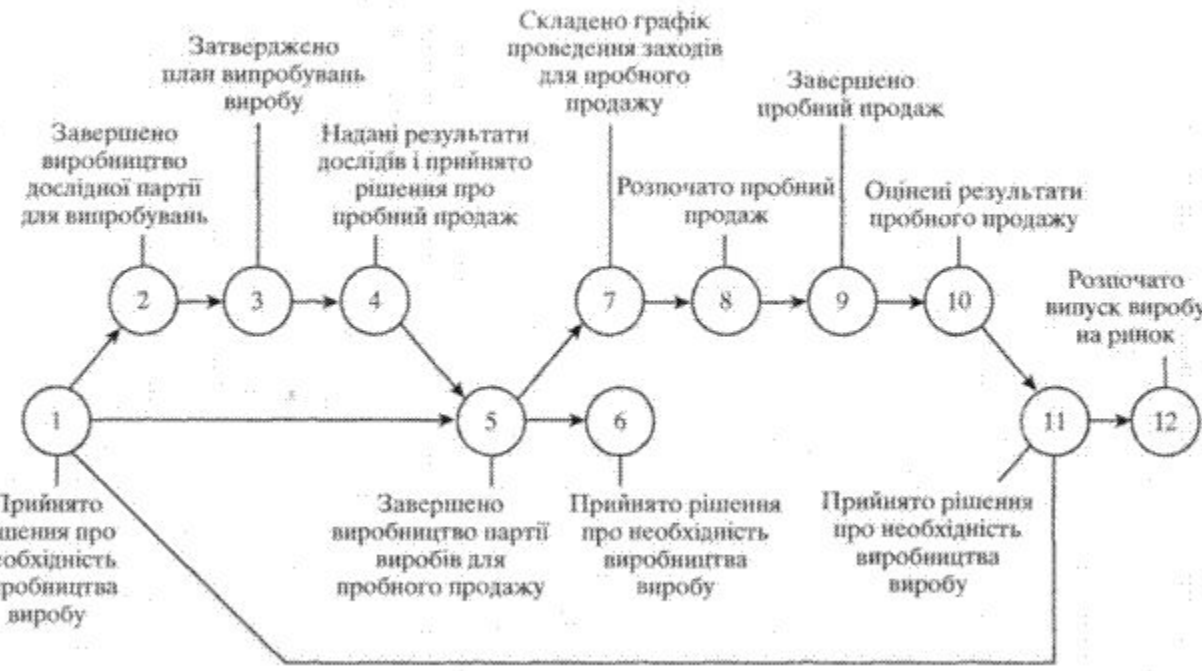


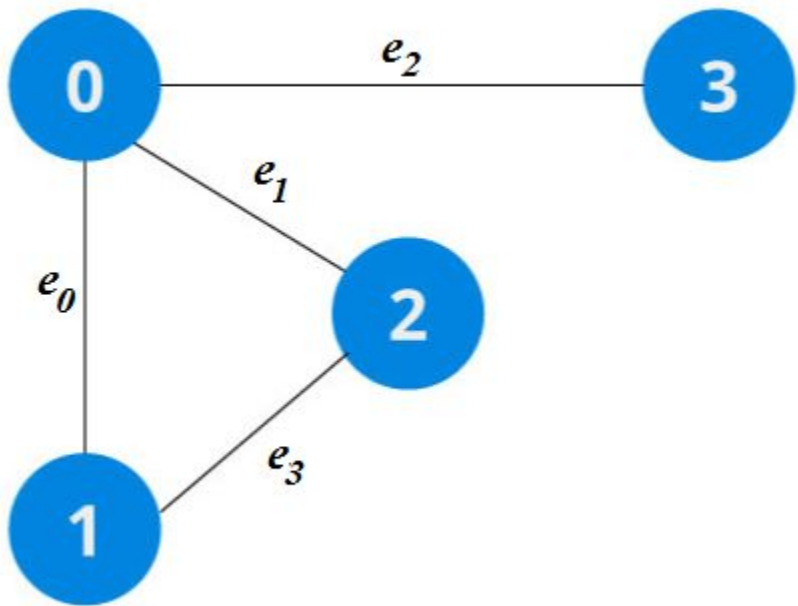
ГРАФ



Київське метро та швидкісний транспорт Kyiv Rapid Transit Map

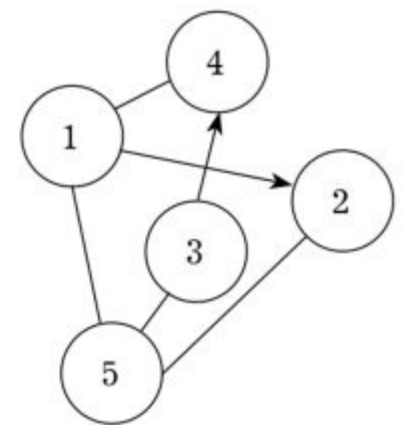
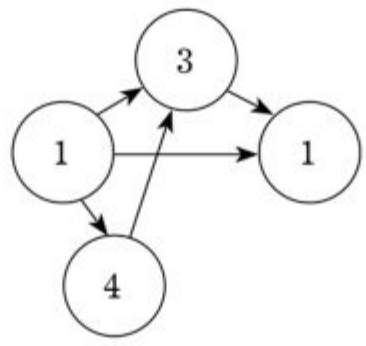
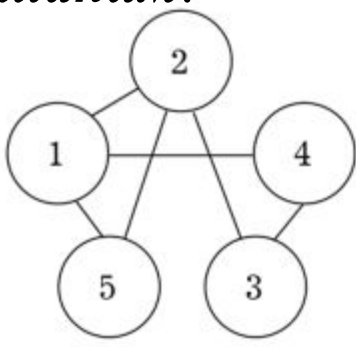


Граф – структура даних, що складається з множини вершин (вузлів або точок) та множини ребер (Edge зв'язків або відрізків).



$V = \{0, 1, 2, 3\}$ від англ. *vertex*
 $E = \{e_0, e_1, e_2, e_3\}$ від англ. *edge*
 $e_0 = (0, 1)$
 $e_1 = (0, 2)$
 $e_2 = (0, 3)$
 $e_3 = (1, 2)$
 $G = \{V, E\}$

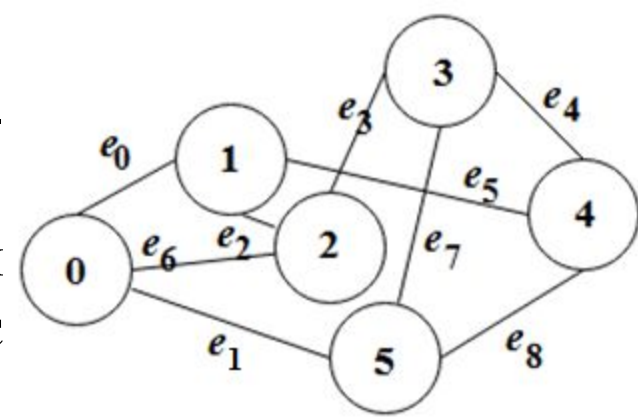
Ребро може мати направленість, тоді воно називається **орієнтоване ребро** або **дуга**. Відповідно і граф може бути **орієнтованим**, **неорієнтованим**, **мішаним**.



Термінологія графів

Суміжність - вершина *суміжна* з іншою вершиною, якщо є ребро, яке їх з'єднує.

Наприклад, вершини 0 та 1 – суміжні, а вершини 2 та 4 не є суміжними, тому що між ними немає ребра.



Інцидентність – ребро інцидентне вершині, якщо вона є його кінцем. Наприклад, ребро e_4 інцидентне вершинам 3 та 4.

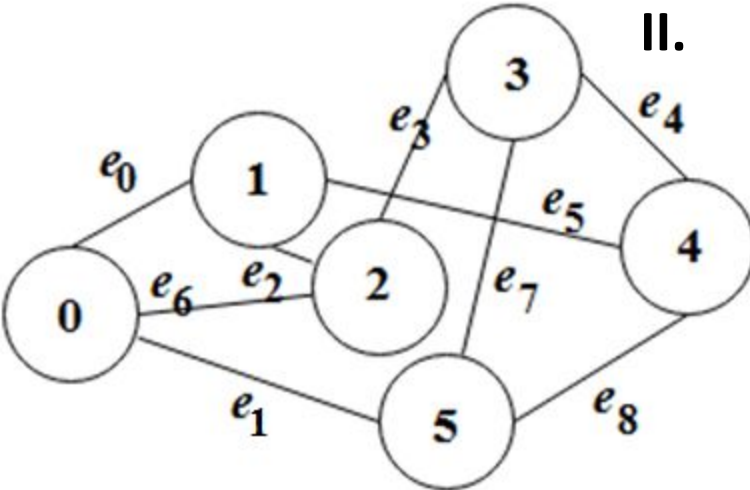
Шлях (Path) - це послідовність об'єднаних ребрами вершин. Наприклад, ми декілька шляхів з вершини 0 до вершини 4: 0-5-4 та 0-2-1-4, 0-1-2-3-5-4.

Якщо шлях не має повторів вершин - це **простий шлях** (Simple Path). Приклад шляху від 0 до 2 з повторами - 0-5-3-4-5-0-2.

Шлях буде **замкненим**, якщо він починається та закінчується в одній вершині. Наприклад, 0-1-2-3-4-5-0.

Графи зазвичай представляють двома способами:

1. **Матриця суміжності** – це двовимірний (2D) масив $A_{V \times V}$ вершин. Кожний рядок і стовпець представляють певну вершину. Якщо значення будь-якого елемента A_{ij} дорівнює 1, це означає, що існує ребро, що з'єднує вершину i вершину j .

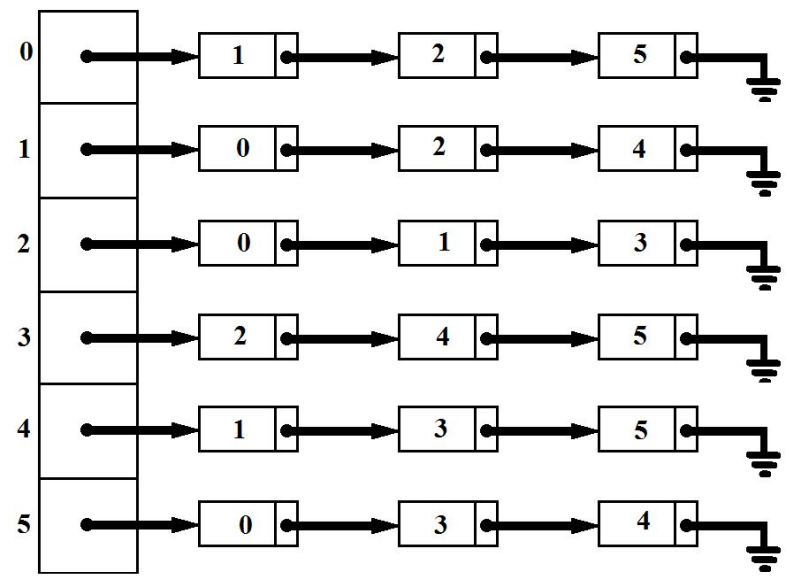


II. Матриця інцидентності – це двовимірний (2D) масив $V_{V \times E}$. Кожний рядок відповідає певній вершині, а стовпець - ребру. Якщо значення будь-якого елемента V_{ij} дорівнює 1, це означає, що ребро j інцидентне вершині i .

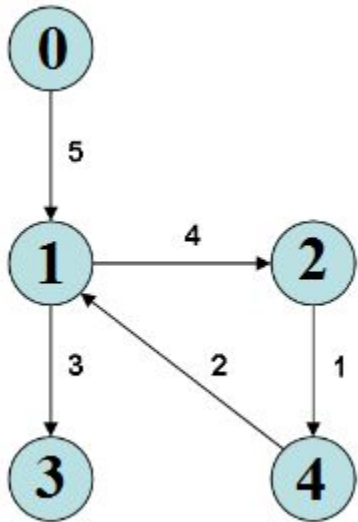
$$A = \begin{matrix} & \begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \end{pmatrix} \end{matrix}$$

$$B = \begin{matrix} & \begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \end{matrix}$$

II. Список суміжності - представлення графу масивом зв'язного списку. Індекс масиву відповідає номеру вершини, а кожний елемент – це голова зв'язного списку із суміжних її вершин.



Граф, у якого кожному ребру (дузі) поставлено у відповідність певне число (вага) називається *зваженим*.



Матриця ваг - це двовимірний (2D) масив $C_{V \times V}$, в якому кожний елемент C_{ij} дорівнює вазі ребра (дузі), що з'єднує вершину i з вершиною j (виходить з вершини i та заходить у вершину j).

$$C = \begin{matrix} & \begin{matrix} 0 & 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{pmatrix} 0 & 5 & 0 & 3 & 0 \\ 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 \end{pmatrix} \end{matrix}$$

Найбільш поширені операції над графами:

- обхід графа;
- додавання/вилучення елементів (вершини, ребра) у граф;
- знаходження шляху від однієї вершини до іншої;
- знаходження оптимального шляху.

Завдання на лабораторну роботу:

Скласти програму, що знаходить вагу всіх замкнених шляхів у зваженому неорієнтованому графі, що проходять через усі вершини графа тільки один раз, та обрати шлях з найменшою вагою.

ДЕРЕВА

До цього ми розглядали переважно тільки лінійні структури даних: список, стек та черга – тобто дані зберігаються послідовно. Ми маємо логічний початок/ голову та кінець структури. Ці структури мають свої певні недоліки та переваги, які ми використовуємо для конкретних цілей у наших програмах. Все залежить від того, які дані треба зберігати та що з ними треба робити.

Хеш-таблиця (англ. *hash* – плутанина) динамічна структура даних, що реалізує інтерфейс асоціативного масиву.

Асоціативний масив або словник (англ. *associative array, dictionary*) – абстрактний тип даних, що дозволяє зберігати дані у вигляді набору пар ”ключ – значення” (*key – value*) та надає доступ до значень за їх ключем.

Основні операції з хеш-таблицями:

- ✓ додавання нової пари ключ-значення;
- ✓ пошуку значення за ключем;
- ✓ видалення пари ключ-значення за ключем.