

УДК 004.021:004.42;519.17

*Байраківська В. В., здобувач вищої освіти,  
Сеник І. О., асистент кафедри  
інформаційних технологій*

## ГРАФИ ТА ЇХ ВИКОРИСТАННЯ В АЛГОРИТМІЗАЦІЇ

*Донецький національний університет імені Василя Стуса, м. Вінниця*

Графи є математичним інструментом для моделювання та розв'язання задач у комп'ютерних науках, інженерії, економіці та інших галузях. Для роботи з елементами в графах використовують зв'язки між вузлами та ребра, що дає змогу ефективного застосування у розв'язанні задач, пов'язаних із пошуком шляхів, маршрутизацією, розумним розподілом ресурсів та іншим. З цією метою графи використовуються в різних індустріях – від соціальних мереж до вантажних перевезень та телекомунікаційних систем. Такі алгоритми розв'язання задач на графах діють і допомагають працювати з даними, представлені у вигляді графів.

Граф є структурою даних, яка визначає спосіб опису зв'язків між набором елементів. Граф складається з набору об'єктів, які називаються вузлами, причому певні пари цих об'єктів з'єднані зв'язками, які називаються ребрами. Говоримо, що два вузли є сусідніми, якщо вони з'єднані ребром. Розрізняють орієнтовний та неорієнтовний графи, мультиграф та псевдограф. У орієнтовного графа ребра називаються дугами [1].

Для ефективного представлення графів у пам'яті комп'ютера використовують два найбільш поширені способи: матриця суміжності та матриця інцидентцій.

Матриця суміжності будується у вигляді таблиці, водночас, якщо в таблиці стоїть 1 – то дві вершини суміжні. В іншому випадку, якщо в таблиця стоїть 0 – то дві вершини між собою не суміжні.

Матриця інцидентцій також будується у вигляді таблиці, водночас у таблиці зображено зв'язки між інцидентними елементами графу (дуга і вершина). Стовпці матриці відповідають дугам, рядки – вершинам. Якщо в таблиці стоїть 1 – то ця дуга є вихідною з цієї вершини. Якщо в таблиці стоїть  $-1$  – то ця дуга є вхідною в цю вершину. В іншому випадку, якщо в таблиця стоїть 0 – то ця дуга не має ніякого зв'язку з цією вершиною.

Поняття графу дає змогу структурувати інформацію та алгоритмізувати цілу низку типових задач. Саме тому існує багато алгоритмів, які працюють з інформацією, що представлена у вигляді графів. Їх ще називають «алгоритми на графах». Зокрема, щоразу, коли ви відкриваєте файл на комп'ютері, останній запускає алгоритм пошуку в пам'яті адреси файла. Під час цього пошук виконується з використанням алгоритму на графах [2].

Розглянемо основні алгоритми на графах. **Пошук в ширину:** алгоритм, що систематично досліджує вершини графу на мінімальній відстані від початкової вершини, знаходить всі досяжні вершини, обчислює відстань до них і створює «дерево в ширину», що охоплює ці вершини. Алгоритм працює як на

орієнтованих, так і на неорієнтованих графах [3]. Використовується в соціальних мережах для виявлення всіх друзів на певній відстані від заданого користувача, у штучному інтелекті для пошукових алгоритмів та у маршрутизації комп'ютерних мереж [4].

**Пошук в глибину:** алгоритм, який намагається йти якомога глибше в графі від початкової вершини, досліджуючи вершини вздовж кожного шляху до кінця. Коли більше немає сусідніх невідвіданих вершин, алгоритм повертається назад до попередньої вершини і продовжує пошук, поки не відвідає всі вершини графу [3]. Використовується в задачах з обмеженнями, у штучному інтелекті для зберігання станів у процесі пошуку та у соціальних мережах для виявлення циклів [5].

**Алгоритм Дейкстри:** знаходить найкоротші шляхи від однієї початкової вершини в орієнтованому графі з невід'ємними вагами ребер. Застосовується у маршрутизації, геолокаційних сервісах та транспортних системах [3].

**Алгоритм Беллмана–Форда** знаходить найкоротші шляхи в графах, де можливі від'ємні ваги ребер і виявляє цикли з від'ємною вагою. Цей алгоритм моделює фінансові мережі, маршрутизацію та транспортну логістику [3].

**Алгоритм Прима** будує мінімальне остовне (кістякове) дерево, додаючи до дерева ребра з найменшою вагою. Пріоритетна черга використовується для швидкого вибору наступного ребра. Використовується для проектування мереж, кластеризації даних та створення електронних схем [3].

**Алгоритм Крускала** будує мінімальне остовне дерево, додаючи найменш важкі ребра, які з'єднують окремі частини графу. Допомогає в оптимізації розташування кабелів, трубопроводів та кластеризації даних [3].

**Метод Форда–Фалкерсона** розв'язує задачу максимального потоку в графі, ітеративно збільшуючи потік. Початковий потік дорівнює нулю, і на кожному кроці знаходиться збільшувальний шлях у резервній мережі, щоб підвищити потік. Є важливим для оптимізації транспорту, ланцюгів постачання та телекомунікаційних мереж [3].

Алгоритми на графах мають широкий спектр застосувань у реальному житті. Вони використовуються в соціальних мережах, мережах передачі даних, логістиці, фінансах і багатьох інших галузях для оптимізації процесів, зменшення витрат і покращення ефективності. Ці алгоритми є основою для багатьох сучасних технологій, включаючи планування маршрутів, аналіз даних та управління мережами.

#### Список використаних джерел

1. Easley D., Kleinberg J. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Book, Cambridge University Press, 2010, 744 p.
2. Кузьменко І. М. *Теорія графів: посібник*. Київ: КПІ ім. Ігоря Сікорського. 2020. 71 с.
3. *Introduction To Algorithms* / T. H. Cormen, Ch. E. Leiserson, R. L. Rivest, C. Stein. The MIT Press Cambridge, Massachusetts London, England. 2009. 1313 p.
4. Sedgewick R., Wayne K. *Algorithms*. Book, Addison-Wesley Professional. 2011. 976 p.
5. Kleinberg J., Tardos É. *Algorithm Design*. Book, Addison-Wesley. 2006. 864 p.