

ВИКОРИСТАННЯ АЛГОРИТМІВ ТОПОЛОГІЧНОГО СОРТУВАННЯ ОРІЄНТОВАНОГО ГРАФУ ПРИ РОЗРОБЛЕННІ РОБОЧОЇ ПРОГРАМИ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ «АЛГОРИТМИ ТА СТРУКТУРИ ДАНИХ»

Навчальна дисципліна «Алгоритми та структури даних» є базовою дисципліною підготовки бакалаврів усіх спеціальностей галузі знань 12 «Інформаційні технології». В рамках цієї дисципліни вивчаються такі фундаментальні питання інформатики, як "алгоритм", "структура даних" та досліджуються загальні принципи побудови та аналізу ефективних алгоритмів. Значна кількість тем тісно пов'язана між собою. Наприклад, перед вивченням алгоритму пірамідального сортування треба обов'язково ознайомити студентів з бінарними деревами, бо піраміда – це алгоритмічна структура даних «бінарне дерево» з певними обмеженнями [1], [2].

На етапі розроблення робочої програми навчальної дисципліни «Алгоритми та структури даних» потрібно сформулювати перелік тем, які будуть вивчатися та проаналізувати взаємозв'язок між ними. Наприклад, «Тема 1. Алгоритми пошуку в рядках», «Тема 2. Алгоритми сортування масивів», «Тема 3. Комбінаторні алгоритми», «Тема 4. Методи оцінки обчислювальної складності алгоритмів», «Тема 5. Алгоритми на графах» і так далі. Очевидно, що алгоритми пошуку потрібно вивчати вже після того, як вивчили алгоритми сортування, бо бінарний пошук в масивах можна застосовувати тільки для упорядкованих масивів. Для наглядної візуалізації отриманого результату доцільно представити його у вигляді ациклічного орієнтованого графу (рис.1).

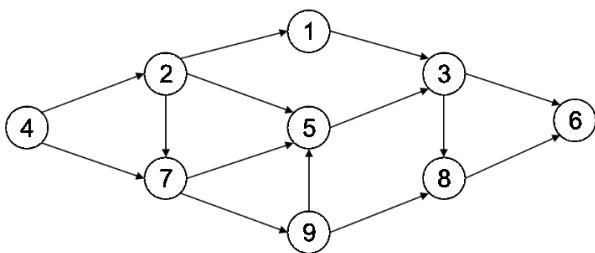


Рис. 1. Ациклічний орієнтований граф

Після цього потрібно розмістити вершини отриманого графу таким чином, щоб на горизонтальній прямій всі його ребра йшли лише зліва направо. Для вирішення цієї задачі доцільно скористатися одним з алгоритмів топологічного сортування [1], [2].

Перший алгоритм полягає у наступному:

1 крок. Шукаємо будь-яку вершину, в яку не входять дуги і нумеруємо її.

2 крок. Помічаємо усі дуги, які виходять із знайденої вершини як такі, що не існують.

3 крок. Повторюємо перші два кроки алгоритму, поки не занумеруємо усі вершини графа.

Але для вирішення цієї задачі існує інший спосіб, який базується на застосуванні алгоритму пошуку в глибину.

Даний алгоритм складається з таких етапів:

1. Починаючи з довільної вершини, проведемо обхід графа за допомогою рекурсивного алгоритму пошуку в глибину.

2. Нумеруємо кожну вершину на зворотному шляху рекурсії, починаючи з максимального номера в порядку спадання номерів.

3. Повторюємо етапи 1 та 2 до тих пір, поки не перенумеруємо усі вершини.

Результатом роботи алгоритму буде граф, вершини якого упорядковані в топологічному порядку (рис. 2).

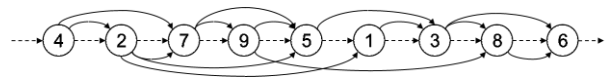


Рис.2. Відсортований граф

Оцінимо обчислювальну складність обох алгоритмів. Перший має обчислювальну складність $O((m+2n)\log(n))$. У другому випадку час роботи алгоритму фактично співпадає з часом обходу графу, тобто має оцінку $O(m+n)$. Таким чином другий алгоритм є більш ефективним, але обидва алгоритми дозволяють побудувати несуперечливу та логічно обґрунтовану послідовність вивчення тем при розробленні робочої програми навчальної дисципліни.

Список літератури

[1] Левитин А. В. *Алгоритми. Введение в разработку и анализ*. Москва, РФ. Вильямс, 2006.

[2] Томас Х. Кормен, Чарльз І. Лейзерсон, Рональд Л. Ривест, и Клиффорд Штайн. *Алгоритмы: построение и анализ*. Москва, РФ. Вильямс, 2013.