Міністерство освіти і науки України

Кременчуцький національний університет   
імені Михайла Остроградського

Навчально-науковий інститут електричної інженерії   
та інформаційних технологій

Кафедра автоматизації та інформаційних систем

НаВчальна дисципліна  
«**аЛГОРИТМИ І СТРУКТУРИ ДАННИХ**»

Звіт

З ПРАКТИЧНОЇ роботи №5

Виконав

студент групи КН-24-1

Озівський В. В.

Перевірив

доцент кафедри КІЕ

Сидоренко В. М.

Кременчук 2025

|  |  |
| --- | --- |
| Тема: | Алгоритми пошуку та їх складність. |
| Мета: | опанувати основні алгоритми сортування та навчитись методам  аналізу їх асимптотичної складності. |

Хід роботи

Постановка завдання

Завдання полягає у розв’язанні задачі, яку потрібно вибрати зі списку.

Номер варіанта відповідає номеру студента у списку групи.

Розв’язання задачі

Задано ациклічний граф:

{1,2,3,4,5,6,7,8,9}{(1,2),(1,3),(2,4),(3,5),(4,5),(4,6),(6,7),(6,9),(7,8),(7,9),(8,9)}.

Побудувати граф і розв’язати задачу топологічного сортування за допомогою алгоритму Кана.

Зображення, що містить ескіз, малюнок, коло, білий

Вміст, створений ШІ, може бути неправильним.

Рисунок 1 – Побудований граф

Розв’язання задачі за допомогою алгоритму Кана.

S = [1], L = []

2. S = [2, 3], L = [1]

3. S = [4, 3], L = [1, 2]

4. S = [3, 5], L = [1, 2, 4]

5. S = [5, 6], L = [1, 2, 4, 3]

6. S = [6], L = [1, 2, 4, 3, 5]

7. S = [7], L = [1, 2, 4, 3, 5, 6]

8. S = [8], L = [1, 2, 4, 3, 5, 6, 7]

9. S = [9], L = [1, 2, 4, 3, 5, 6, 7, 8]

10. S = [], L = [1, 2, 4, 3, 5, 6, 7, 8, 9]

Отже, результуюче топологічне сортування: 𝐿 = [1, 2, 4, 3, 5, 6, 7, 8, 9]

Відповіді на контрольні питання

1. Які переваги і недоліки алгоритму Кана порівняно з алгоритмом DFS для топологічного сортування графа?

Алгоритм Кана зручний для виявлення циклів і добре підходить для потокової обробки. Недолік – потребує додаткових структур (черга, масив ступенів).

DFS простіший у реалізації та не потребує додаткових структур, але складніший у виявленні циклів і потребує обробки після завершення обходу.

1. Яка складність часу і пам’яті для кожного з алгоритмів у найгіршому і найкращому випадках?

Для обох алгоритмів часова складність у найгіршому і найкращому випадках – O(V + E).

Алгоритм Кана використовує додаткову пам’ять для зберігання ступенів вершин і черги (O(V)).

DFS використовує пам’ять для позначення відвіданих вершин і стек викликів (O(V)).

1. Чи можна застосовувати алгоритм Кана до графів з вагами на ребрах? Як це порівняти з DFS?

Так, обидва можуть працювати з графами, де є ваги на ребрах, але ігнорують самі ваги, бо топологічне сортування не враховує їх — важливий лише напрям.

1. Як впливає структура графа на швидкість роботи кожного з цих алгоритмів?

У графах з великою кількістю зв’язків (густих) обидва працюють ефективно.

У графах з довгими ланцюгами DFS може бути ефективнішим через менші витрати на структури. Алгоритм Кана вимагає більше обчислень для ступенів.

1. Чи є обмеження використання кожного алгоритму для певних типів графів або завдань?

Обидва алгоритми працюють тільки з ациклічними орієнтованими графами (DAG).

Кан не працює без попередньо обчислених ступенів входу. DFS потребує обережності з порядком додавання у вихід.

1. Які варіанти оптимізації можна застосувати для кожного алгоритму з метою поліпшення його продуктивності

Для Кана можна використати пріоритетну чергу для контролю порядку сортування або ефективніше зберігати ступені.

Для DFS — можна уникнути рекурсії через ітеративну реалізацію або використовувати бітові масиви для visited.