

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМ. ІГОРЯ СІКОРСЬКОГО»

# ПРОГРАМНІ ЗАСОБИ ВПОРЯДКУВАННЯ ГРАФУ ДИДАКТИЧНОЇ ОНТОЛОГІЇ

8.05010302 – Інженерія програмного забезпечення

**КОПИЛОВА ВІКТОРІЯ ЮРІЇВНА**

**НАУКОВИЙ КЕРІВНИК:**

**К.Т.Н., ДОЦ. ТИТЕНКО СЕРГІЙ ВОЛОДИМИРОВИЧ**

# МЕТА ТА ОБ'ЄКТ ДОСЛІДЖЕННЯ

## Мета роботи:

Оптимізація алгоритму впорядкування графу дидактичної онтології, на основі топологічного сортування.

## Задачі дослідження:

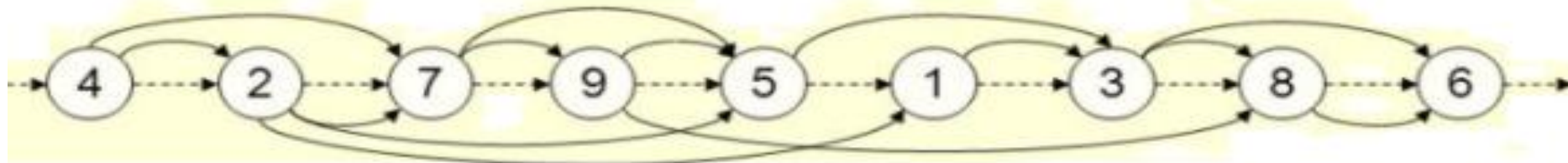
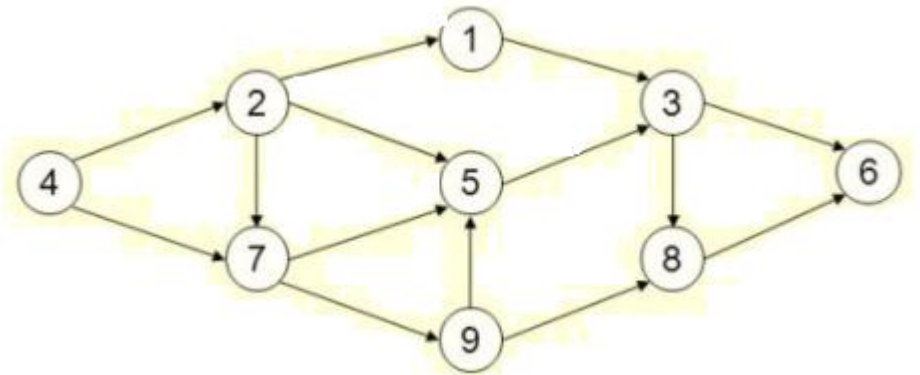
1. Проаналізувати існуючі методи впорядкування зважених орієнтованих графів.
2. Дослідити метод виділення семантичного ядра та удосконалити його алгоритмом топологічного сортування.
3. Програмно реалізувати оптимізований алгоритм впорядкування орієнтованого зваженого графу.
4. Провести порівняльний аналіз сортування графу декількома методами.

## Об'єкт дослідження:

Програмні системи впорядкування зважених орієнтованих графів.

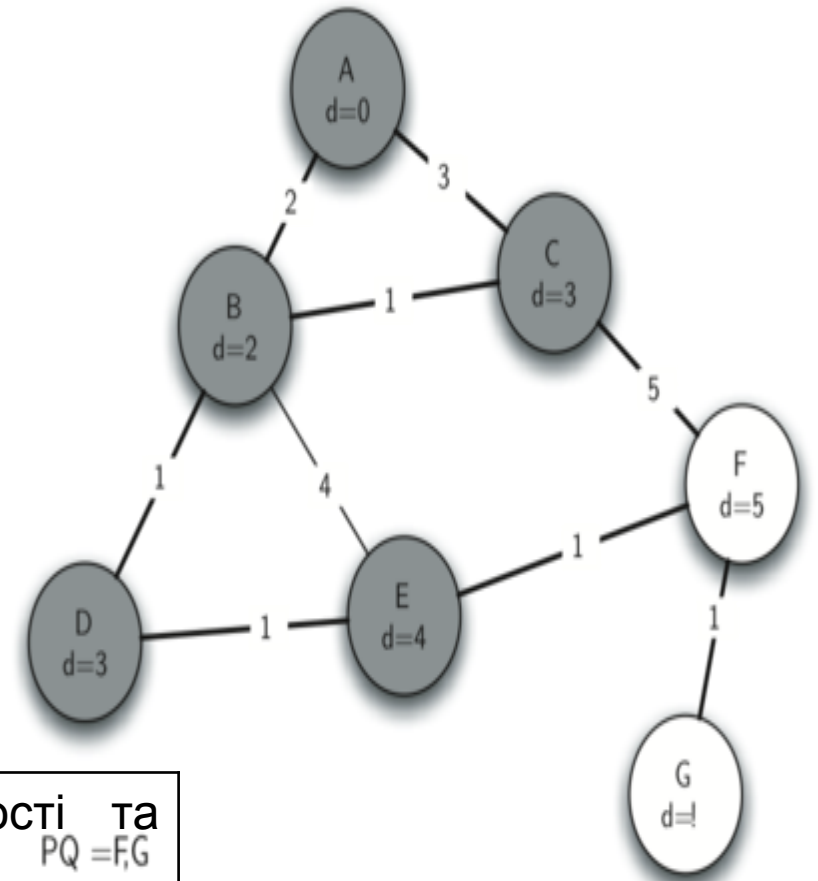
# ТОПОЛОГІЧНЕ ВПОРЯДКУВАННЯ ГРАФА

Топологічне впорядкування орієнтованого графа – це лінійне впорядкування його вершин таким чином, що для кожного направлено ребра  $G = \{u,v\}$  з вершини  $u$  до вершини  $v$ , таким чином що вершина  $u$  передуює вершині  $v$ .



# АЛГОРИТМ ПРИМА

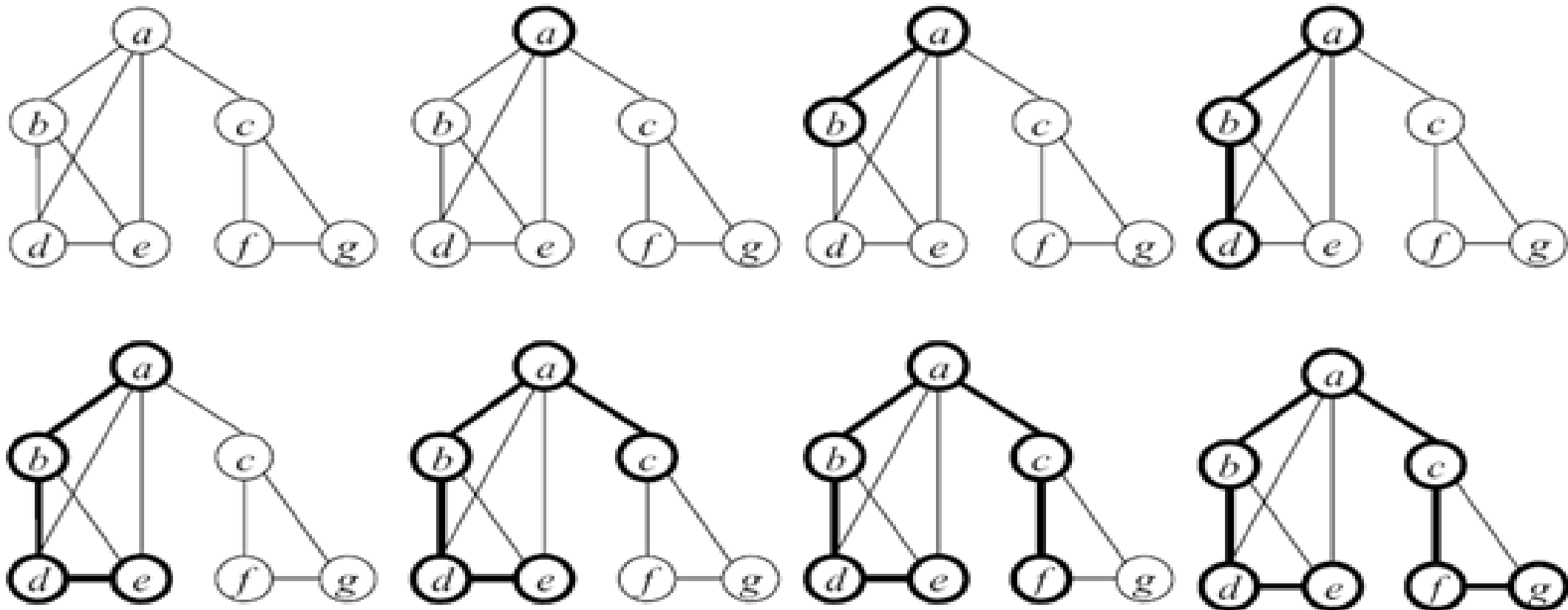
Алгоритм будує мінімальне остовне дерево методом перебору вершин та обираючи з них ребро з мінімальною вартістю.



Матриця суміжності	Списки суміжності та двійкова купа	Списки суміжності та купа Фібаначі $PQ = F, G$
$O(n^2)$	$O( ( V + E )*\log V  )$	$O(  E + V *\log V  )$

# АЛГОРИТМ ПОШУКУ ВГЛИБ

Асимптотична складність -  $O(V+E)$



# АЛГОРИТМ ВИДІЛЕННЯ СЕМАНТИЧНОГО ЯДРА

**Семантичний центр** - поняття найсильніше зв'язане з іншими поняттями курсу.

**Семантичне ядро** - набір ключових понять, які визначаються на основі семантичного центру.

Сутність алгоритму полягає у наступному:

Перед включенням до послідовності поточного поняття, воно перевіряється на наявність передуючих понять, якщо такі є, то спочатку відбувається включення до послідовності кожного з цих понять;

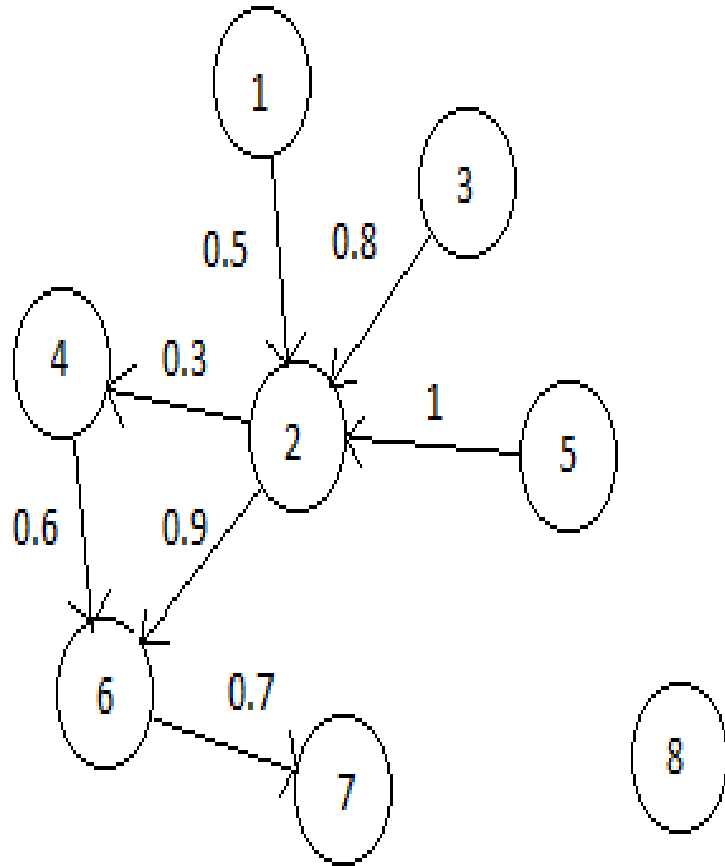
Потім включається поточне поняття, а після цього в чергу додаються поняття, що слідує за ним.

Такий процес застосовується для кожного поняття під час його включення до послідовності.

# АНАЛІЗ ІСНУЮЧИХ АЛГОРИТМІВ

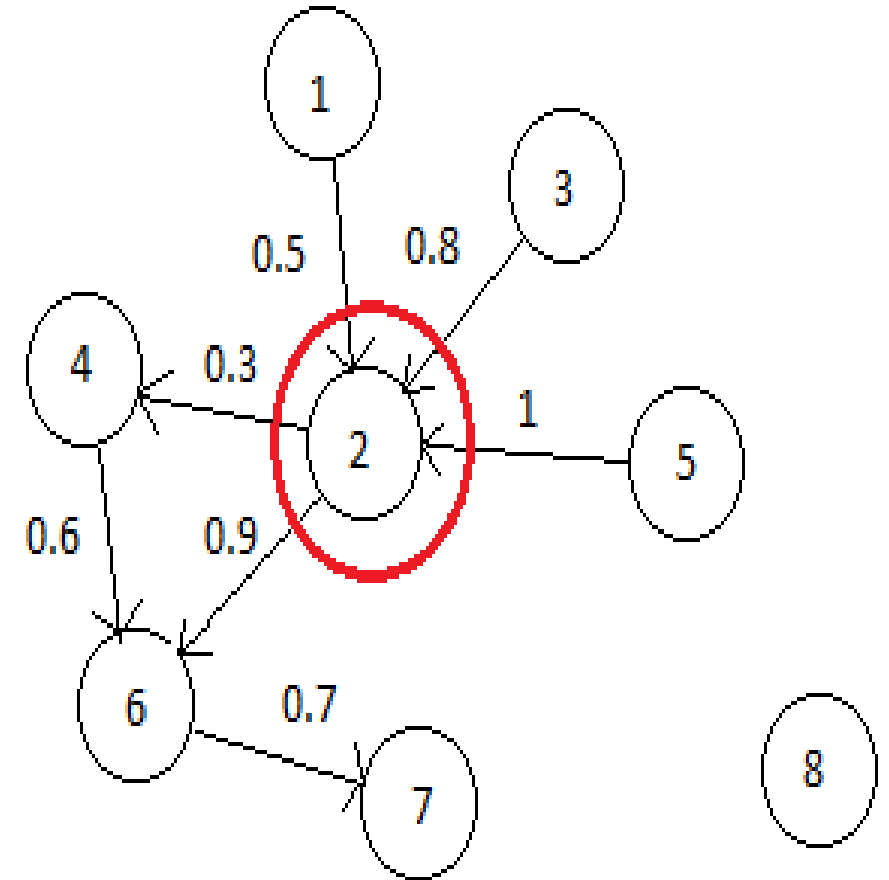
	Алгоритми топологічного сортування	Алгоритм виділення семантичного ядра
Недоліки	<ol style="list-style-type: none"><li>1. Не враховує особливості обраної предметної області, під час перебору не зважають на ступінь зв'язності вершин</li></ol>	<ol style="list-style-type: none"><li>1. Потребує багато ресурсів (пам'яті) та часу виконання.</li><li>2. Не можна використовувати в інших задачах.</li></ol>
Переваги	<ol style="list-style-type: none"><li>1. Можуть бути використані в багатьох прикладних задачах.</li><li>2. Асимптотична складність варіюється до <math>O(n)</math>.</li></ol>	<ol style="list-style-type: none"><li>1. Враховує особливості дидактичної онтології.</li></ol>

# ОПТИМІЗАЦІЯ ГРАФУ ДИДАКТИЧНОЇ ОНТОЛОГІЇ



1. Знайдемо семантичне ядро графу дидактичної онтології, а саме вершину, в яку входить найбільша кількість ребр.

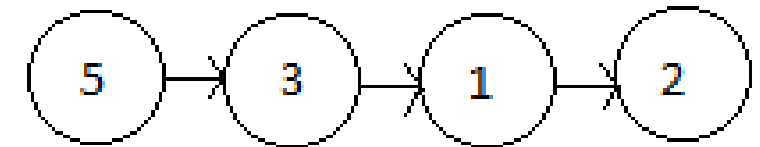
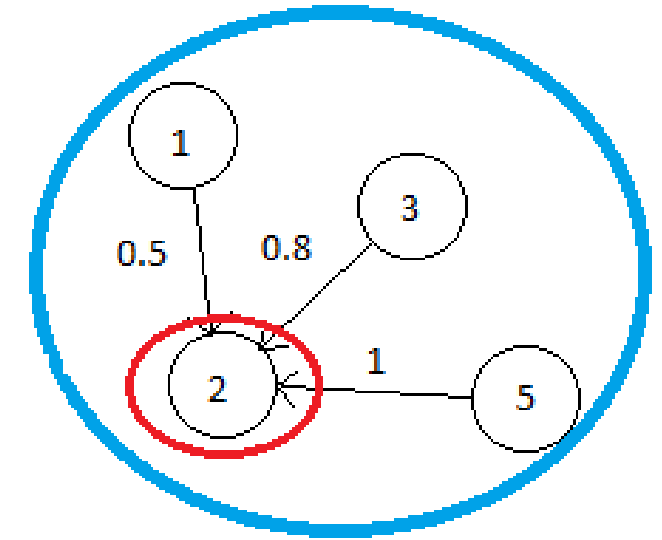
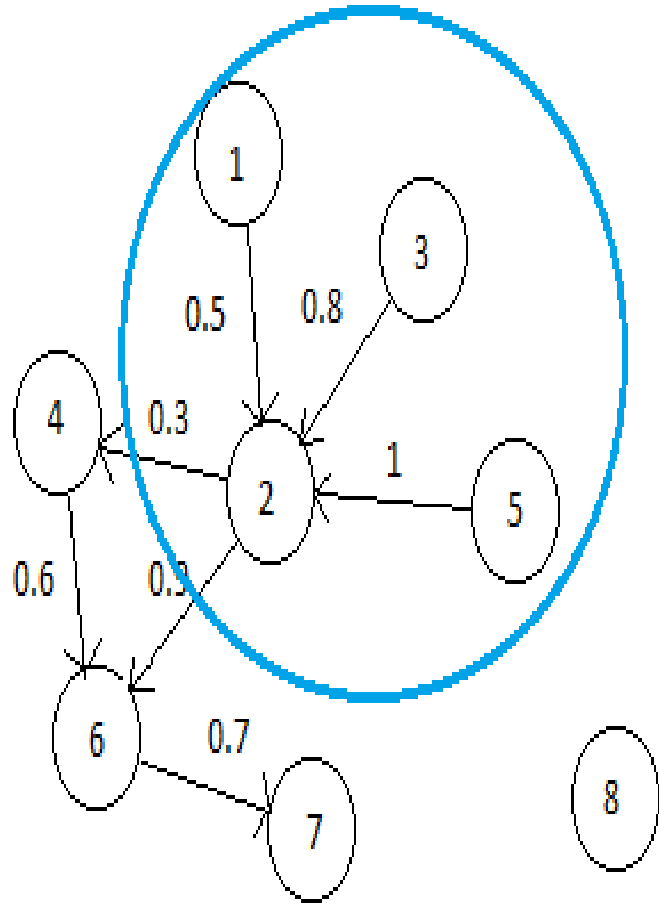
2. Задаємо результуючу множину семантичне ядро.





# ВИДІЛЕННЯ МНОЖИНИ, ПЕРЕДУЮЧОЇ СЕМАНТИЧНОМУ ЯДРУ

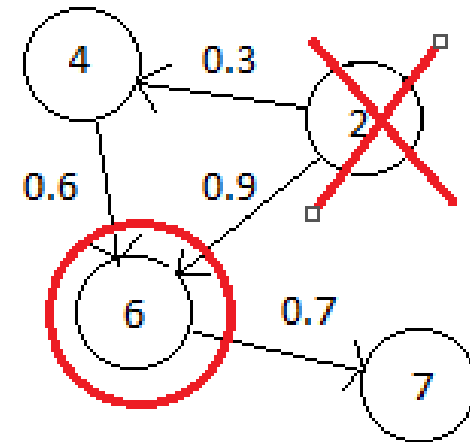
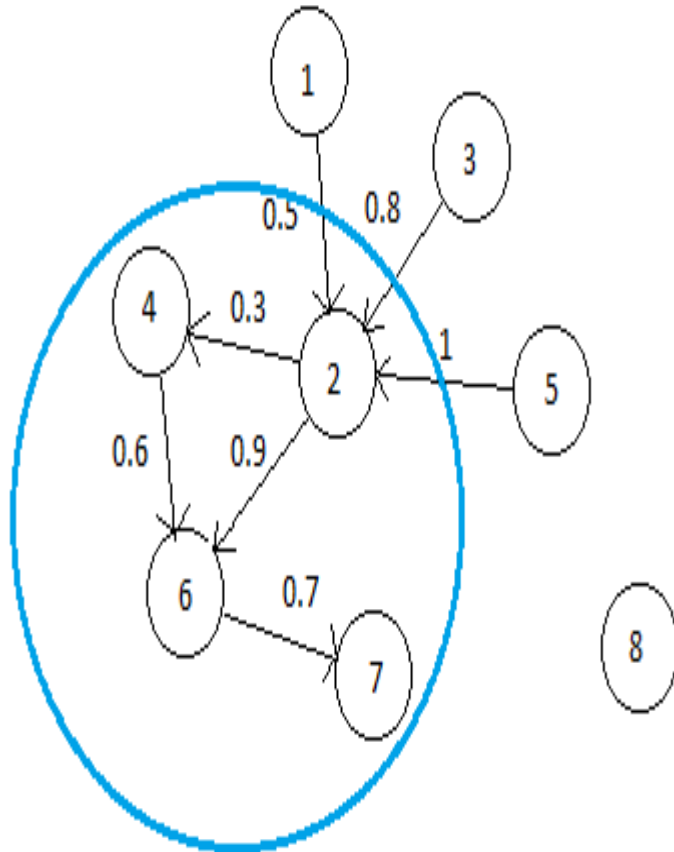
3. Знайдемо множину понять-вершин графу, які передують семантичному ядру.
4. Знайдений підграф відсортуємо методом топологічного впорядкування вглиб та відповідно до ваги ребр.
5. Додаємо до результуючої множини відсортований підграф.



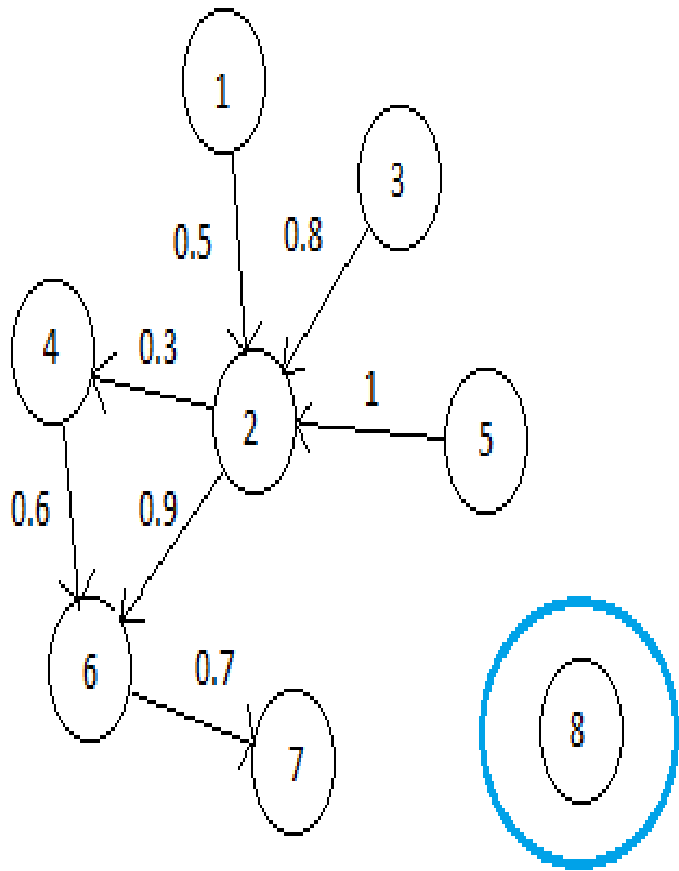
# ВИДІЛЕННЯ МНОЖИНИ, СЛІДУЮЧОЇ ЗА СЕМАНТИЧНИМ ЯДРОМ

5. Знайдемо множину понять-вершин графу, що розташовані після семантичного ядра.

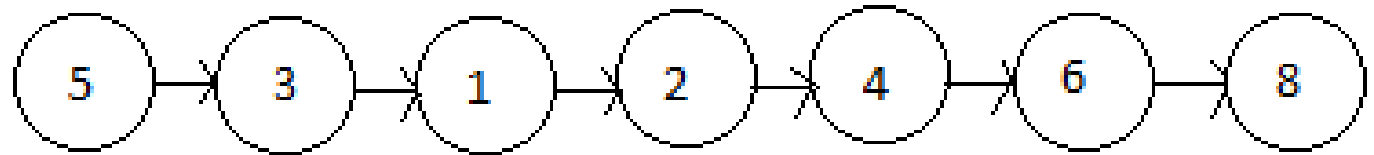
7. На отриманому підграфі виділимо ядро підграфу та рекурсивно виконати пункти 3-5



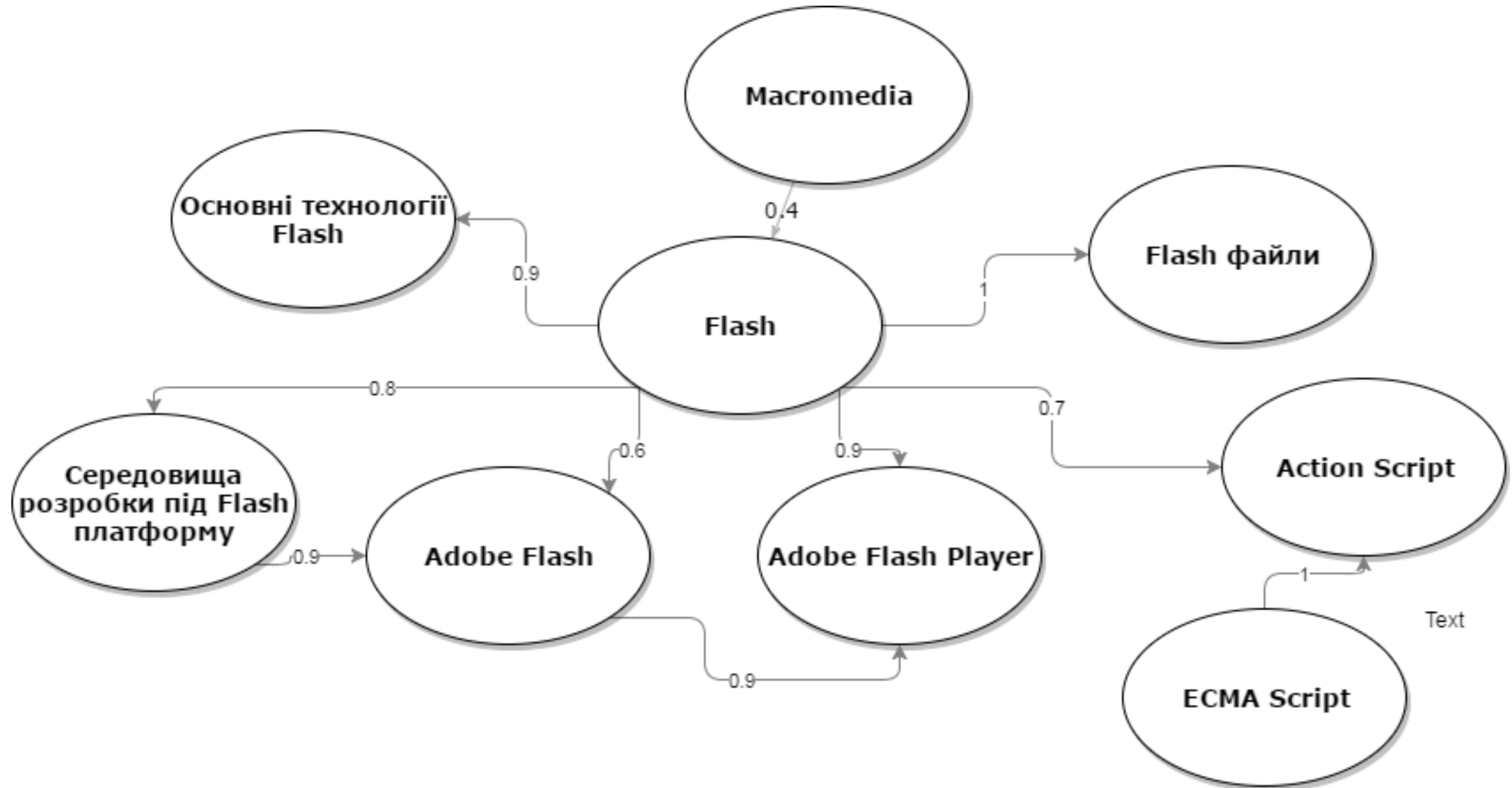
# ЗАКІНЧЕННЯ АЛГОРИТМУ



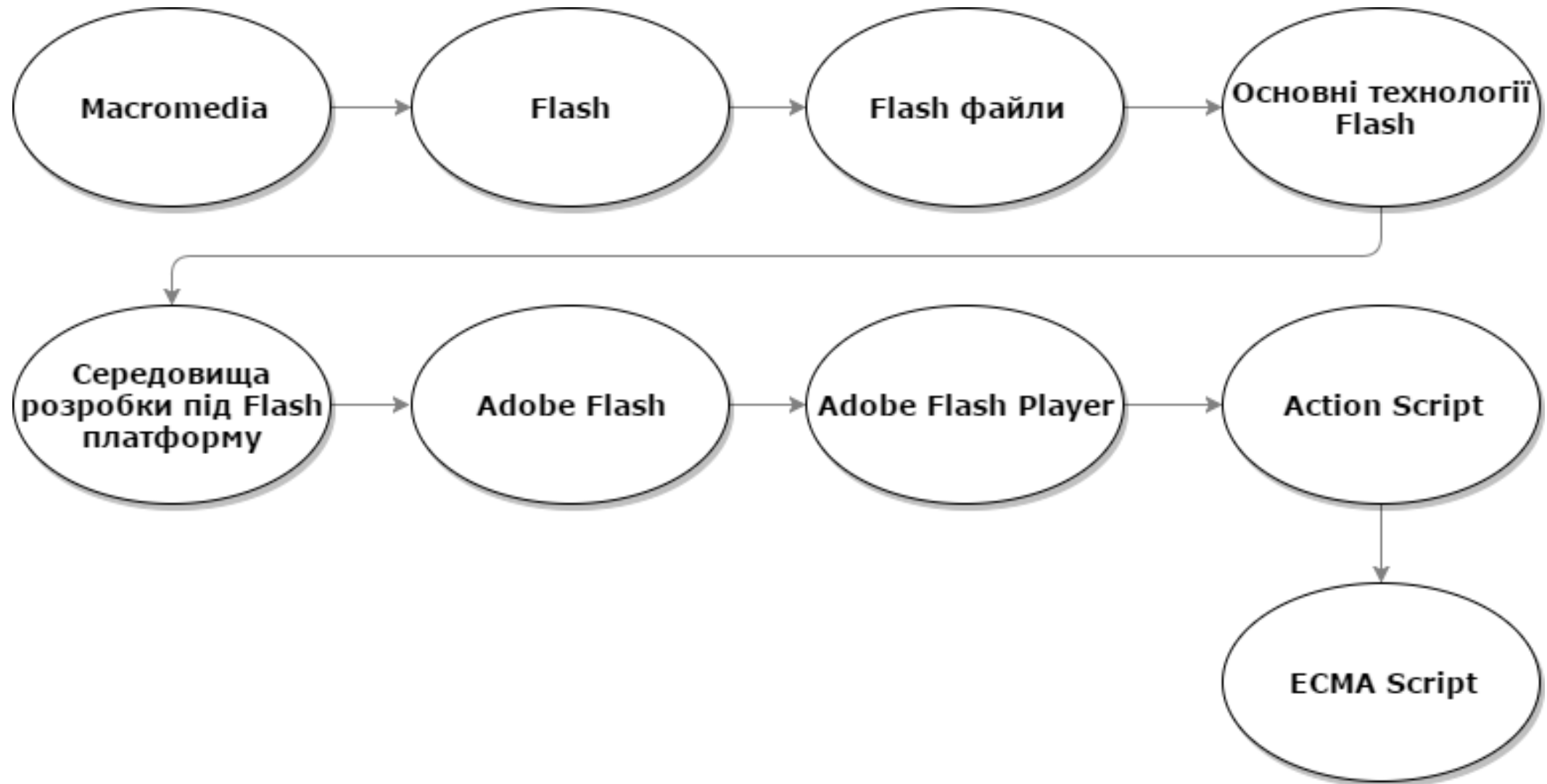
8. Перевірити чи відповідає розмір послідовності кількості цільових зв'язаних понять. Якщо ні, по черзі запустити пункти 1-7 для зв'язаних понять, що досі не в послідовності.
9. Додати до послідовності незв'язані поняття



# ПРИКЛАД РОБОТИ НА ГРАФІ «FLASH»



# ПРИКЛАД РОБОТИ НА ГРАФІ «FLASH»

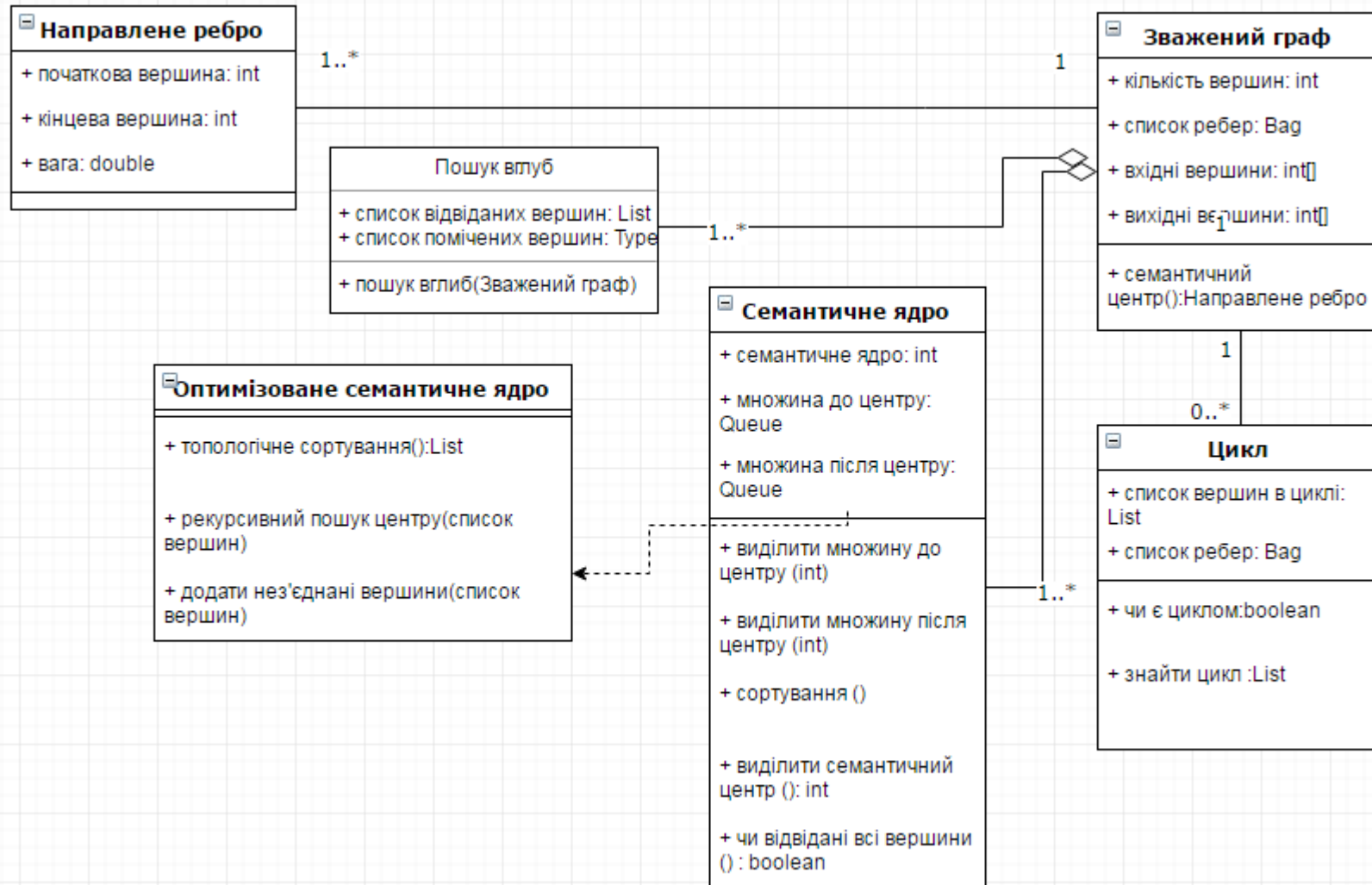


# ПОРІВНЯЛЬНИЙ АНАЛІЗ АЛГОРИТМІВ СОРТУВАННЯ ГРАФУ ДИДАКТИЧНОЇ ОНТОЛОГІЇ «FLASH»

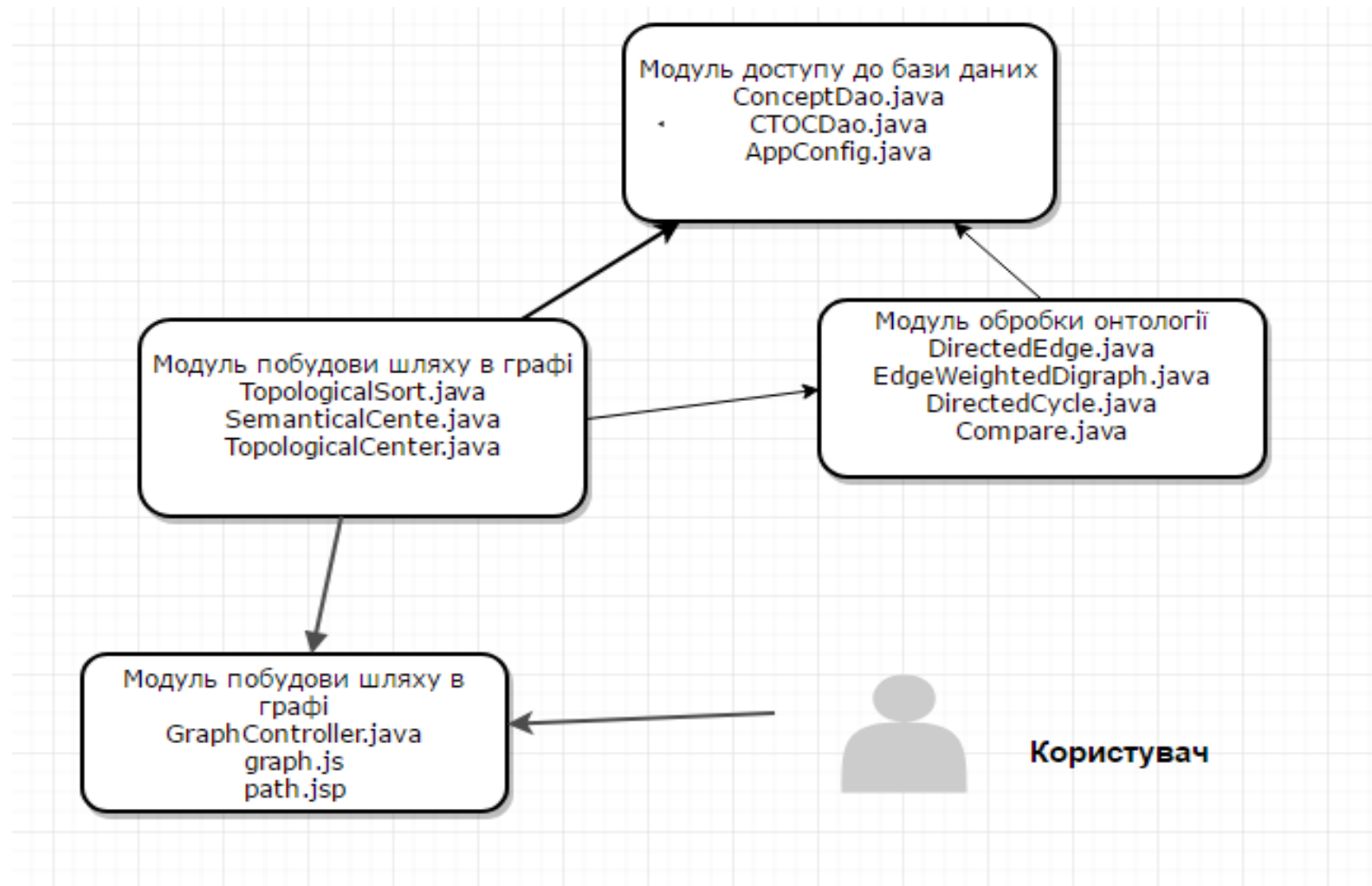
Топологічне сортування вглиб	Виділення семантичного ядра	Виділення семантичного ядра з застосуванням топологічного сортування
669 789 нано секунд	1 000 881 нано секунд	968 718 нано секунд

Топологічне сортування	Виділення семантичного ядра
Основні технології Flash	Flash файли
Середовища розробки під Flash платформу	Основні технології Flash
Adobe Flash	Середовища розробки під Flash платформу
Adobe Flash Player	Adobe Flash
Action Script	Adobe Flash Player
ECMA Script	Action Script
Flash файли	ECMA Script

# ДІАГРАМА КЛАСІВ В ПОРЯДКУВАННЯ ГРАФУ ДИДАКТИЧНОЇ ОНТОЛОГІЇ



# МОДУЛЬНА СТРУКТУРА ПРОГРАМНОГО КОМПЛЕКСУ





# ВИСНОВКИ

В ході роботи було розв'язано задачу оптимізації впорядкування графу дидактичної онтології.

Робота має такі теоретичні та практичні результати:

1. Здійснено аналіз існуючих алгоритмів топологічного сортування зважених орієнтованих графів. Проаналізовано алгоритм виділення семантичного ядра.. Показано, що алгоритми топологічного сортування є найбільш оптимальними для графів, але не підходять для вирішення задачі впорядкування графу дидактичної онтології тому що не враховують особливості предметної області, алгоритм виділення семантичного ядра було розроблено з урахуванням усіх особливостей предметної області, але він потребує багато ресурсів для зберігання проміжних множин.
2. Запропоновано оптимізувати алгоритм виділення семантичного ядра топологічним сортуванням. Показано, що час роботи оптимізованого алгоритму менше, кількість використаних ресурсів зменшено.
3. Розроблено програмну систему побудови шляху в графі дидактичної онтології, яка виконує сортування графу декількома методами, приводить порівняння часу роботи алгоритмів, побудованих шляхів

**Дякую за увагу!**