УДК 004.932.4

Трофімук А.А., Семенюк В.Я.
Луцький національний технічний університет

ВІЗУАЛЬНИЙ РЕДАКТОР ГРАФІВ

В даній науковій роботі обґрунтовується актуальність графів та показується, як можна розв'язувати задачі з їх допомогою. Метою мого дослідження стала розробка програмного забезпечення для візуальної роботи з графами.

Ключові слова: *GAlgorithms, алгоритм Дейкстри, граф, графовий редактор, застосування графів, під-граф, теорія графів.*

Введення

В даній роботі обґрунтовується актуальність графів та показується, як можна розв'язувати задачі з їх допомогою. Для цих цілей доцільно використовувати програмне забезпечення спеціального призначення, але склалося так, що на даний час в світі немає такого, яке було б варте Вашої уваги. Одні програми не є візуальними, що без суперечок – дуже погано, адже граф, це складова геометричних фігур, якщо ми їх бачимо, то краще розуміємо поставлену перед собою задачу та її розв'язок. Візуалізація також є невід'ємною частиною при моделюванні дорожнього маршруту, маршрутизації в комп'ютерних мережах, в розв'язуванні задач з фізики та хімії. З цього випливає, що це є основна частина того, чим повинне володіти сучасне програмне забезпечення такого рівня. Але це лише одна сторона медалі, якщо дивитися ширше, то можна побачити в цих програмах ще один недолік, вони погано справляються із поставленими задачами, які містять в собі велику кількість вхідних даних, що також є не маловажним фактором. Враховуючи всі ці аспекти, метою мого дослідження стала розробка програмного забезпечення гідного рівня.

Перша робота по теорії графів, що належить відомому швейцарському математику Л. Ейлеру, з'явилася в 1736 році. Спочатку теорія графів здавалася досить незначним розділом математики, тому що вона мала справу в основному з математичними головоломками. Проте подальший розвиток математики і особливо її напрямків дав сильний поштовх для розвитку теорії графів. Вже в XIX сторіччі, графи використовувалися при побудові багатьох моделей. А тепер теорія графів використовується в фізиці, для побудови схем розв'язання задач, за допомогою графів значно спрощується розв'язання задач з електротехніки. Архітектори використовують графи для найбільш раціонального розміщення об'єктів і прокладання доріг на плані забудови населеного пункту. Біологи використовують графи для розв'язання задач з генетики. Навіть на математичних заняттях учні та студенти використовують графи для розв'язання геометричних задач та задач практичного змісту. Математичні головоломки також залишаються частиною теорії графів, особливо якщо віднести до них знамениту проблему чотирьох фарб, що інтригує математиків і до цього дня. Взагалі, останнім часом дослідження в областях, що традиційно відносяться до дискретної математики, займають все більш помітне місце. Поряд з такими класичними розділами математики, як математичний аналіз, диференціальні рівняння, у навчальних планах спеціальності "Прикладна математика" і багатьох інших спеціальностей, з'явилися розділи по математичній логіці, алгебрі, комбінаториці і теорії графів. З цього випливає, що актуальність даної наукової роботи є дуже високою.

Так що ж таке граф?

Граф – це пара множин V і E, елементи множини V називаються вершинами (англійською Vertex), множина E містить впорядковані та невпорядковані пари вершин. Невпорядкована пара вершин називається ребром, впорядкована – дугою. В таблиці 1, продемонстровані типи графів.

Таблиця 1

Типи графів

Тип графа	***Ребра***	***Кратні ребра***
Простий граф	Неорієнтовані	Ні
Мультиграф	Неорієнтовані	Так
Орієнтований граф	Орієнтовані	Ні
Орієнтований мультиграф	Орієнтовані	Так

Візуальний редактор графів "GAlgorithms"

Перейдемо до основної мети мого дослідження, це розробка власного програмного забезпечення для роботи з графами, яке називається "GAlgorithms". Розробляючи дану програму, я врахував всі мінуси та плюси вже існуючого програмного забезпечення. До основних плюсів та нововведень моєї розробки можна віднести: візуальний редактор та вміння справлятися із задачами, які містять в собі велику кількістю вхідних даних, удосконалений алгоритм Дейкстри для пошуку найкоротшого маршруту в графі (тепер він працює набагато швидше) та розроблений власний алгоритм пошуку під-графів даного графа.

Удосконалений алгоритм Дейкстри

Алгоритм Дейкстри – це алгоритм на графах, для знаходження найкоротшого маршруту від однієї з вершин графа до всіх інших. Алгоритм широко застосовується в програмуванні і технологіях, наприклад, його використовує протокол OSPF для усунення кільцевих маршрутів.

Формулювання завдання:

Дана мережа автомобільних доріг, що з'єднує міста Волинської області. Деякі дороги односторонні. Знайти найкоротший маршрут від Луцька до кожного міста області (рухатися можна тільки по дорогах).

Формальне визначення:

Даний орієнтований граф $G=(V,E)$ *без петель і дуг негативної ваги. Знайти найкоротші маршрути від деякої вершини графа* G *до всіх інших вершин цього графа.*

Мною була проведена оптимізація алгоритму Дейкстри, це робилося для того, щоб він працював швидше та справлявся із задачами, які мають в собі велику кількість вхідних даних. Умовні позначення для мого алгоритму продемонстровано в таблиці 2, а псевдокод цього алгоритму в таблиці 3.

Таблиця 2

Умовні позначення

Позначення	***Опис***
V	Множина вершин графа
E	Множина ребер графа
$w[i,j]$	Вага (довжина) ребра i,j
A	Початкова вершина
U	Множина вершин, які були вже відвідані
$d[u]$	По закінченню роботи алгоритму, дорівнює довжині найкоротшого маршруту від вершини A до вершини u
$p[u]$	По закінченню роботи алгоритму, містить найкоротший маршрут від вершини A до вершини u

Таблиця 3

Псевдокод

Присвоїмо $d[a]\leftarrow 0,\ p[a]\leftarrow a$
Для всіх $u\in V$ відмінних від A
Присвоїмо $d[u]\leftarrow\infty$
Поки $\exists v\notin U$
Нехай $v\in U$ - вершина з мінімальним $d[v]$
Для всіх $u\notin U$ таких, що $v,u\in E$
Якщо $d[u]>d[v]+w[v,u]$ то
Змінимо $d[u]\leftarrow d[v]+w[v,u]$
Змінимо $p[u]\leftarrow p[v],u$

Продемонструю роботу алгоритму на практиці. Відкриваємо в програмі "GAlgorithms" карту України та будуємо шляхи від міста до міста, нехай це будуть наші дороги, після цього знаходимо найкоротший маршрут від Луцька до Севастополя. Даний пошук продемонстрований на Рис. 1.

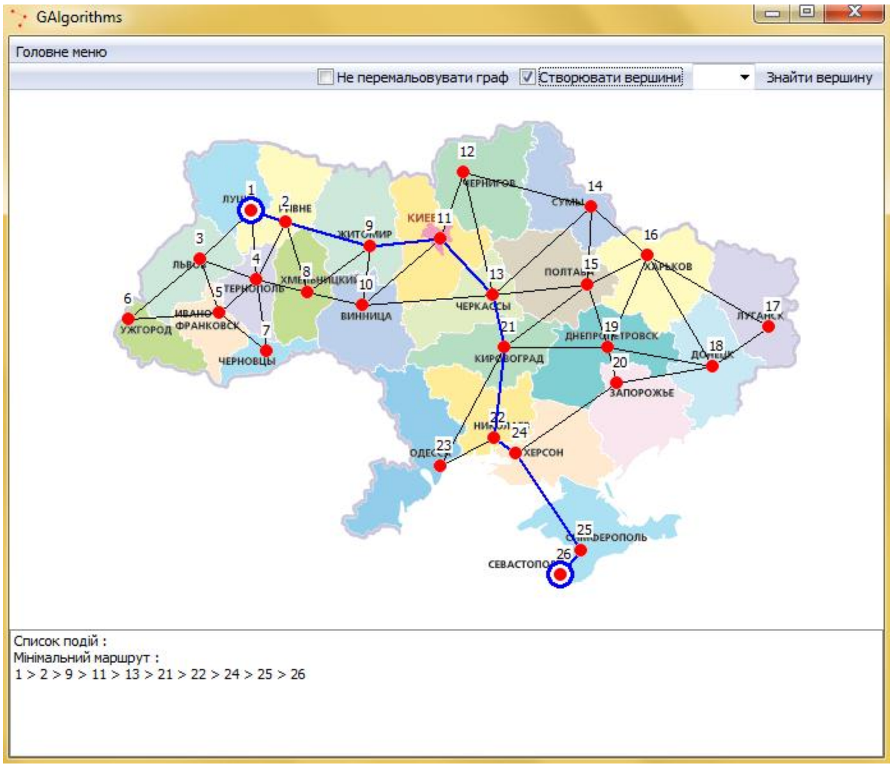

Рис. 1 – знаходження найкоротшого маршруту від міста до міста на карті України

Пошук під-графів даного графа

Задача про знаходження під-графів даного графа відноситься до класу NP-повних задач. Вперше вона була сформульована в 1972 році Ричардом Карпом.

Замітка:

В теорії алгоритмів, класом NP (від англійського Non-Deterministic Polynomial) називають алгоритми, час роботи яких, суттєво залежить від розміру вхідних даних.

Визначення:

Даний граф $G=(X,A)$, *де* $X=\{x_i\}, i=1,2,...,n$ *– множина вершин,* $A=\{a_i\}, i=1,2,...,m$ *– множина дуг. Під-графом* $G'=(X',A')$ *даного графа* G, *називається такий граф* G', *для якого виконуються умови* $X'\subseteq X$ *та* $A'\subseteq A$. *Приклад вихідного графа наведено на Рис. 2, а його під-графи на Рис. 3.*

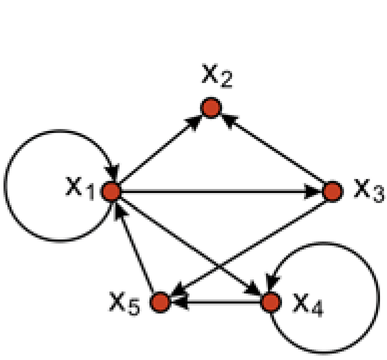

Рис. 2 – вихідний граф

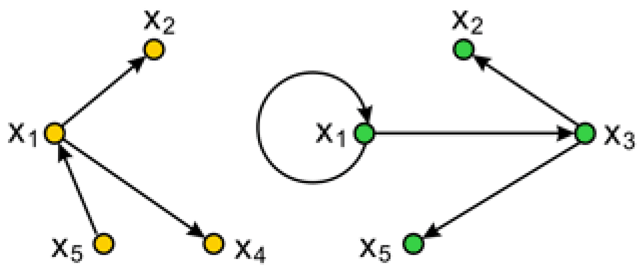

Рис. 3 – під-графи вихідного графа

Як і для будь-яких NP-повних задач, ефективного алгоритму для пошуку під-графів – не існує. Повний перебір всіх можливих під-графів розміру K, з перевіркою того, чи є хоча б один з них повним – неефективний, оскільки повне число таких під-графів з V вершинами дорівнює біноміальному коефіцієнту $\binom{V}{K} = \frac{V!}{K!(V-K)!}$.

Саме тому, я вирішив розробити оптимальний алгоритм для пошуку під-графів. Даний алгоритм я вирішив реалізувати на базі алгоритму Брона-Кербоша (алгоритм пошуку всіх клік графа). Алгоритм використовує той факт, що всяка кліка в графові є його максимальним по включенню повним під-графом. Починаючи з одиночної вершини (утворюючої повний під-граф), алгоритм на кожному кроці намагається збільшити вже побудований повний під-граф, додаючи до нього вершини з множини кандидатів. Висока швидкість забезпечується відсіканням при переборі варіантів, які свідомо не приведуть до побудови кліки, для чого використовується додаткова множина, в якій містяться вершини, які вже були використані для збільшення повного під-графа. Умовні позначення для мого алгоритму продемонстровано в таблиці 4, а псевдокод цього алгоритму в таблиці 5.

Таблиця 4

Умовні позначення

Позначення	***Опис***
$Temp$	Множина, що на кожному кроці рекурсії містить повний під-граф для даного кроку
$Candidates$	Множина вершин, які можуть поповнити $Temp$
Not	Множина вершин, які вже використовувалися для поповнення $Temp$ на попередніх кроках

Таблиця 5

Псевдокод

Процедура Find ($Candidates$, Not):
Поки $Candidates <> 0$ і Not не містить вершини, з'єднаної зі всіма вершинами з $Candidates$
    Вибираємо вершину V з $Candidates$ і додаємо її в $Temp$
    Формуємо множину $NCandidates$ і $NNot$, видаляючи з $Candidates$ і Not вершини не з'єднані з V
    Якщо $NCandidates = 0$ і $NNot = 0$ то
        $Temp$ - під-граф
    Інакше
        Викликаємо Find ($NCandidates$, $NNot$)
    Видаляємо V з $Temp$ і $Candidates$ та поміщаємо в Not

Продемонструю роботу алгоритму на практиці. В програмі "GAlgorithms" створимо необхідну кількість вершин та з'єднаємо їх, після цього знайдемо під-графи даного графа. Даний пошук продемонстрований на Рис. 4.

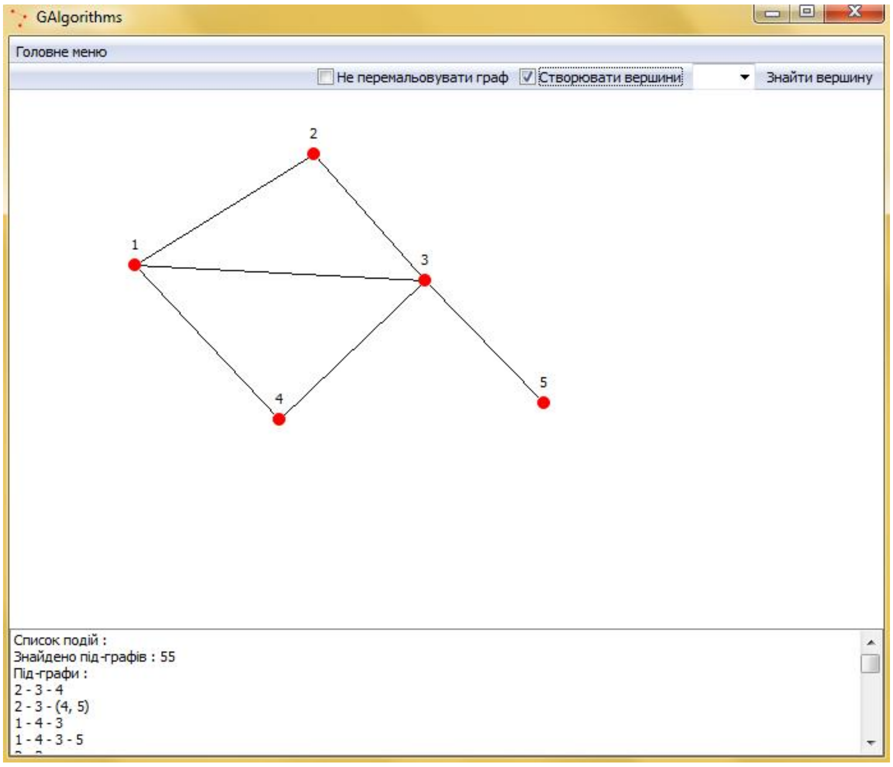

Рис. 4 – знаходження під-графів даного графа

Я продемонстрував Вам лиш основні плюси моєї розробки, тобто ті, які мають наукову новизну. На цьому можливості "GAlgorithms" не вичерпуються, також, в програмі реалізовані такі алгоритми: знаходження Ейлеревого циклу та шляху, знаходження найкоротшого маршруту, знаходження остового дерева, знаходження під-графів даного графа і т.д. Також, для приємної роботи із програмою, реалізоване збереження результатів в файл, відкриття результатів з файлу, збереження графа як малюнка, задавання графа матрицею суміжності, випадкове розміщення вершин і їх з'єднання, знаходження вершин по їхньому номеру, переміщення вершин, видалення вершин або ребер та багато чого іншого. Для практичного застосування програми, в ній є можливість відкриття зображення з файлу, таке зображення може бути, наприклад, картою країни, картою доріг, схемою комп'ютерних мереж. Після чого, наприклад, на карті країни можна знайти найкоротший маршрут через міста, від початкового міста до кінцевого, на схемі комп'ютерної мережі легко візуалізувати оптимальне прокладання кабелю та інше. Програмний інтерфейс та код реалізовано на базі об'єктно-орієнтованої мови програмування Delphi, в сучасному засобі розробки програмного забезпечення CodeGear RAD Studio Delphi 2007 із використанням стандартних компонентів.

Висновки

Дана наукова робота доводить, що дискретна математика, поряд з такими класичними розділами математики, як математичний аналіз, диференціальні рівняння, різні розділи по математичній логіці, алгебрі, комбінаториці і теорії графів тісно пов'язані із сучасним програмуванням. З цього зрозуміло, що дана тема є актуальною і перспективною в наш час. А саме головне це те, що дані методи, які описувались, можуть допомогти розв'язати деякі життєві проблеми.

Список використаної літератури

1. В. А. Евстигнеев, Применение теории графов в программировании, Наука, 1985 год, 352 страницы.
2. Д. Андерсон, Дискретная математика и комбинаторика, Вильямс, 2004 год, 958 страниц.
3. К. Берж, Теория графов и ее применения, Издательство иностранной литературы, 1962 год, 320 страниц.
4. Р. Уилсон, Введение в теорию графов, Мир, 1977 год, 208 страниц.
5. У. Татт, Теория графов, Мир, 1988 год, 424 страницы.