

Кушнерьов О.С.
Сумський державний педагогічний
університет ім. А.С.Макаренка

АЛГОРИТМІЧНО РОЗВ'ЯЗНІ ЗАДАЧІ ТЕОРІЇ ГРАФІВ

Науковий керівник: кандидат фіз.-мат. наук, доцент Лукашова Т.Д.

Сучасне навчання з необхідністю передбачає використання інформаційних технологій при вивченні будь-яких дисциплін. Особливо це стосується математики, яка власними підходами, формалізованою мовою та методами дає змогу розв'язувати велику кількість задач сьогодення, але разом з тим є досить складною для сприйняття пересічним студентам. Для полегшення розуміння специфічного навчального матеріалу створюється велика кількість програмних засобів, серед яких наразі варто виділити системи комп'ютерної математики, в яких закладені потужні алгоритми розв'язування типових задач.

Велику кількість реальних ситуацій зручно моделювати за допомогою рисунків, що складаються з точок (вершин), які представляють основні елементи ситуації, та ліній (ребер), що з'єднують певні пари цих вершин і описують зв'язки між ними. Такі рисунки у математичній літературі називають графами.

З графами, самі того не помічаючи, ми стикаємося постійно. Найпростішим прикладом графів є схема доріг чи ліній метрополітену. У ній точкам відповідають населені пункти (станції метро), а лініям, що їх сполучають, – шляхи руху автомобілів (поїздів). Досліджуючи свій родовід, ми будуємо так зване генеалогічне древо. І це древо також є прикладом графа (до речі, такі графи називають кореневими деревами).

Графи є зручним засобом опису зв'язків між об'єктами. Зокрема, розглядаючи граф, що зображує мережу доріг між населеними пунктами, можна визначити маршрут проїзду від пункту А до пункту Б. Якщо таких маршрутів виявиться кілька, можна вибрати з них оптимальний, наприклад, найкоротший або найбезпечніший. Для вирішення подібних завдань вибору

потрібно проводити певні дослідження графів й мати певні алгоритми їх обробки. На сьогоднішній день існує досить велика кількість готових алгоритмів дослідження графів, які моделюють найрізноманітніші завдання з багатьох сфер людської діяльності.

Метою даної статті є огляд деяких прикладних задач теорії графів та алгоритмів їх розв'язання.

Алгоритм Крускала є одним з алгоритмів побудови мінімального кістякового дерева зваженого неорієнтованого графа. Алгоритм було вперше описано Джозефом Крускалом у 1956 році. Алгоритм Крускала розпочинається з побудови виродженого лісу, що містить V дерев, кожне з яких складається з однієї вершини. Далі виконуються операції об'єднання двох таких дерев, для чого використовуються найкоротші можливі ребра, поки не утвориться єдине дерево. Це дерево і буде мінімальним кістяковим деревом.

Інший відомий оптимізаційний алгоритм – алгоритм Дейкстри – є алгоритмом пошуку найкоротших шляхів від однієї вершини до будь-якої іншої у зваженому орграфі. Класичний алгоритм Дейкстри працює тільки для графів, що не містять дуг від'ємної довжини, та широко використовується у програмуванні та комунікаційних технологіях, біоінформатиці, базах даних, інженерії, зокрема, його використовує протокол OSPF для усунення кільцевих маршрутів.

Для розв'язування задачі пошуку найкоротших шляхів у графах з від'ємними вагами ребер може використовуватись алгоритм Беллмана-Форда. Цей алгоритм заснований на ідеях динамічного програмування. Коротко ідею динамічного програмування можна описати наступним чином. Початкова задача розбивається на менші підзадачі, які розв'язуються рекурсивно. Далі розв'язок початкової задачі отримується із розв'язків підзадач. При цьому кожна з окремих підзадач може розв'язуватись більше ніж один раз. Для того щоб уникнути зайвої роботи в алгоритмах динамічного програмування вводяться спеціальні таблиці (матриці), які зберігають значення уже розв'язаних задач. Через це такі алгоритми вимагають значних ресурсів пам'яті для збереження відповідних значень.

Розглянемо задачу пошуку найкоротшого шляху між кожною парою вершин у графі. Звичайно, цю загальнішу задачу можна розв'язати багатократним застосуванням алгоритмів Дейкстри або Беллмана-Форда з послідовним вибором кожної вершини графу як початкової. Проте, є прямий спосіб розв'язування цієї задачі за допомогою алгоритму Флойда-Уоршола. У ньому довжини дуг можуть бути від'ємними, проте не повинно бути циклів із від'ємною довжиною.

Алгоритм Джонсона дозволяє знайти найкоротші шляхи між усіма парами вершин зваженого орієнтованого графу. Цей алгоритм працює, якщо у графі містяться ребра з додатними чи від'ємними вагами, але відсутні від'ємні цикли. Ідея алгоритму Джонсона полягає в наступному. Уявимо собі, що граф не містить ребер з від'ємними вагами. Тоді для знаходження найкоротших відстаней між усіма парами вершин ми можемо застосувати п

разів алгоритм Дейкстри. У випадку, якщо в графі присутні ребра від'ємної ваги, можна змінити вагу ребер таким чином, щоб вага кожного ребра стала додатною і тоді знову застосувати алгоритм Дейкстри n разів.

Сучасний навчальний процес неможливий без використання інформаційних систем, які підтримують дистанційні технології навчання, мобільне та електронне навчання, а також звичайні лекційні, семінарські і лабораторні заняття. Створюється велика кількість програмних засобів підтримки, які можна не тільки безпосередньо використовувати під час навчання, а й організовувати самостійну роботу, дослідницьку діяльність, поліпшувати наочність моделювання процесів тощо.

У наш час існує велика група програм, що підтримують вивчення дискретної математики, серед яких виділимо: *MaxFlow*, *Графоаналізатор*, *GraphBuilder*, *Maple*, *GraphMaker*.

Програмний продукт *MaxFlow* призначений для наочного вивчення і застосування деяких алгоритмів на орієнтованих графах:

- 1) пошук максимального потоку в мережі;
- 2) пошук компонент сильної зв'язності;
- 3) пошук оптимального шляху (найкоротшого з шляхів з максимальним потоком);
- 4) пошук мінімального основного дерева;
- 5) пошук всіх найкоротших шляхів у графі (з виділенням шляху, вибраного користувачем).

Програма *Графоаналізатор* дозволяє створювати граф, використовуючи матриці суміжності або візуальний метод, застосовувати різні алгоритми опрацювання від пошуку найкоротшого шляху до перевірки на планарність. У програмі передбачено збереження графа або його зображення.

Програма *GraphBuilder* візуалізує найбільш популярні алгоритми на графах – Дейкстри, Флойда, Прима і Крускала.

Програма *GraphMaker* призначена для побудови і розрахунку мережових графів: візуалізація графа, побудова критичного шляху і резервів часу тощо.

Пакет символічної математики *Maple*, окрім стандартних команд, які задають граф та характеризують певні його властивості, передбачає можливість написання власного алгоритму розв'язання задачі про найкоротшу відстань.

Задача знаходження найкоротших шляхів відноситься до оптимізаційних проблем дискретної математики і може бути розв'язана повним перебором усіх варіантів. Проте, починаючи з середини XX століття існують більш економні методи, які реалізуються багатьма програмними засобами. Аналіз математичних програмних засобів, таких як *MaxFlow*, *Графоаналізатор*, *GraphBuilder*, *Maple* тощо, показує, що більшість середовищ вимагає додаткового часу на вивчення інтерфейсу і команд пакету, а типові алгоритми втаємничені або написані незнайомою мовою програмування, що не завжди дає змогу уявити ідею розв'язання. З іншого

боку, самостійне відтворення алгоритму у програмних середовищах не тільки спонукає до подальших пошуків, а і підвищує рівень одержаних знань та вмінь.

Саме тому, хоч і є напрацьовані алгоритми розв'язування задачі на відшукування найменшої відстані, ми поставили задачу виділити з цих алгоритмів найпростіший для сприйняття та написати програму його візуальної підтримки (рис.1). Також була поставлена задача на написання коду візуальної підтримки алгоритму Крускала на відшукування мінімального остовного дерева (рис.2).

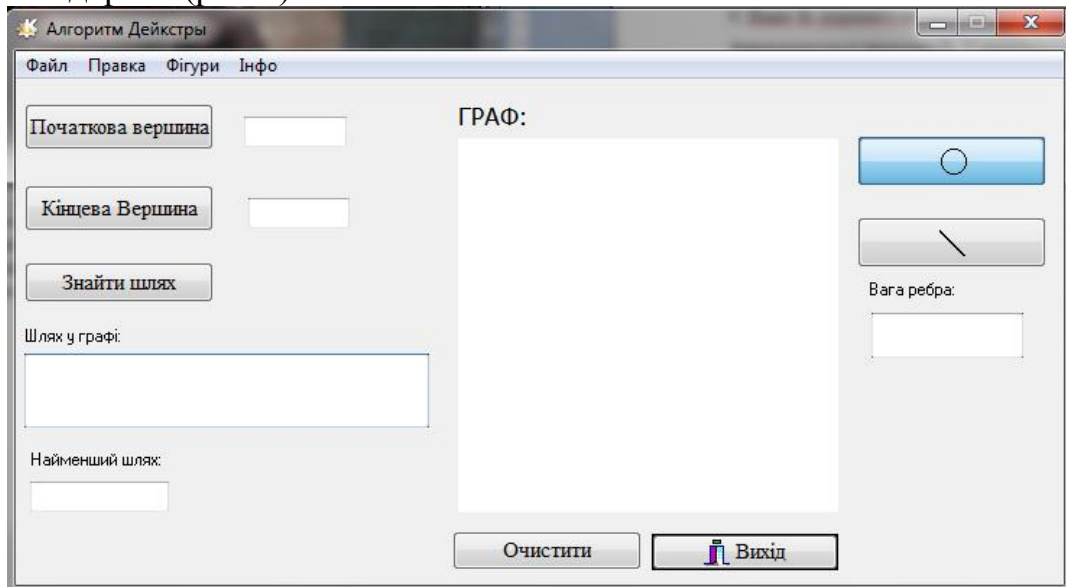


Рис.1. Фрагмент авторської програми «Алгоритм Дейкстри»

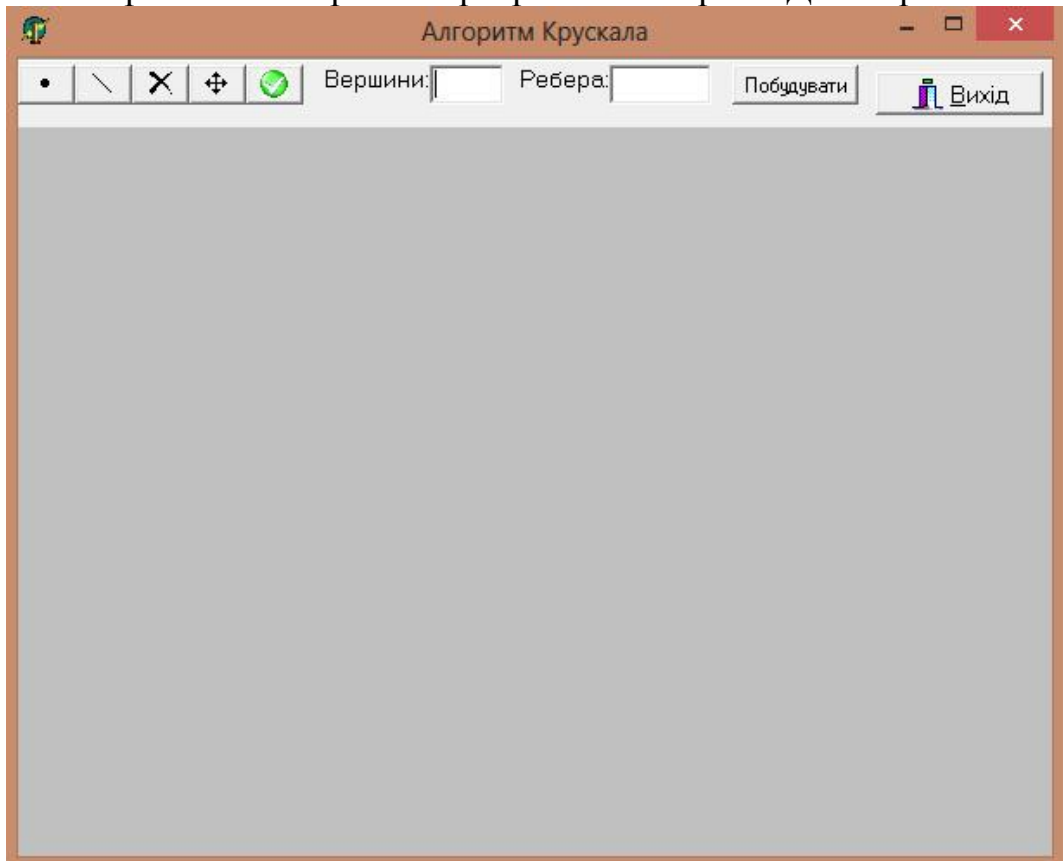


Рис.2. Фрагмент авторської програми «Алгоритм Крускала»

Апробація авторської програми «Алгоритм Дейкстри» була здійснена на заняттях з дискретної математики в Сумському державному педагогічному університеті ім. А.С. Макаренка у 2011-2013 роках. Використання програми підвищило не тільки зацікавленість темою, а і якість засвоєння навчального матеріалу.

ЛІТЕРАТУРА

1. Алгоритм Беллмана – Форда / [електронний ресурс]. – режим доступу: <http://urban-sanjoo.narod.ru/bellman-ford.html>
2. Алгоритм Дейкстри / [електронний ресурс]. – режим доступу: <http://ru.wikipedia.org/wiki>
3. Алгоритм Джонсона / [електронний ресурс]. – режим доступу: <http://uk.wikipedia.org/wiki>
4. Алгоритм Крускала / [електронний ресурс]. – режим доступу: <http://uk.wikipedia.org/wiki>
5. Бобровский С. Delphi 7. Учебный курс. - СПб.: Питер, 2004. - 735 с.
6. Алгоритм Флойда – Уоршола / [електронний ресурс]. – режим доступу: <http://uk.wikipedia.org/wiki>
7. Применение MAPLE для изучения теории графов / [електронний ресурс]. – режим доступу: <http://rudocs.exdat.com/docs/index-229878.html#7527500>
8. Примеры применения пакета networks / [електронний ресурс]. – режим доступу: <http://mapleseven.net/Glava16/Index14.htm>
9. Теория графов / [електронний ресурс]. – режим доступу: <http://www.ict.edu.ru/ft/004708/maple30.html>