

*Левченко М.Р., студентка 1 курсу
Спеціальності 122 «Комп'ютерні науки»
Науковий керівник:
Гончар В. М., асистент кафедри
Інформаційних технологій*

АЛГОРИТМИ ЗНАХОДЖЕННЯ НАЙКОРОТШОГО ШЛЯХУ В ОРІЄНТОВАНИХ ГРАФАХ

Донецький національний університет імені Василя Стуса, м. Вінниця

У сучасному світі, де інформаційні потоки швидко розповсюджуються і глобальна комунікація стає все більш важливою, знаходження найкоротшого шляху між точками стає актуальною задачею в багатьох областях. Велику роль у вирішенні цих задач відіграють класичні алгоритми, такі як алгоритм Дейкстри, алгоритм Флойда-Уоршелла та алгоритм Беллмана-Форда.

1.1 Класичні алгоритми знаходження найкоротшого шляху

- Дейкстра
- Флойда-Уоршелла
- Беллмана-Форда

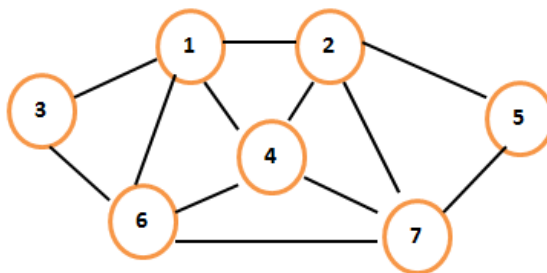


Рисунок 1. Приклад неорієнтованого графа

Алгоритм голландського вченого Едсгера Дейкстри знаходить всі найкоротші шляхи з однієї наперед заданої вершини графа до всіх інших.

Використовуючи цей алгоритм, за наявності всієї необхідної інформації, є можливість, наприклад, дізнатися яку послідовність доріг краще використовувати, щоб дістатися з одного міста до іншого.

Приклади використання алгоритму:

1. Є карта міста на якій нанесені точки розміщення лікарень. Знайти до кожного будинку найближчу лікарню [1].

2. Дана мережа велосипедних доріг у Вінницькій області. Знайти найкоротшу відстань від Вінниці до кожного міста області, рухаючись лише дорогами.

3. Є мережа велосипедних доріжок міста Вінниці. Знайти найкоротший шлях від кожного будинку до торгового центру «Sky Park»

2) Алгоритм Флойда-Уоршола (Floyd-Warshall algorithm) Цей метод порівнює всі можливі шляхи між кожною парою вершин у графі, враховуючи всі можливі проміжні вершини. Для кожної трійки вершин (i, j, k) , де i, j і k представляють вершини графа, а n є загальною кількістю вершин, алгоритм порівнює поточну оцінку шляху між вершинами (i, j) з сумою оцінок шляхів між вершинами (i, k) і (k, j) . Якщо сума оцінок є меншою за поточну оцінку шляху між вершинами (i, j) , то оцінка оновлюється і стає меншою.

Цей процес повторюється для всіх можливих комбінацій вершин (i, j, k) , змінюючи значення k від 1 до n , ітеруючи по всіх вершинах графа. Це дозволяє послідовно покращувати оцінки найкоротшого шляху між будь-якими двома вершинами у графі. Часова складність цього алгоритму становить $O(n^3)$ порівнянь [2].

Приклади використання алгоритму:

1. Навігаційні програми на смартфонах використовують цей алгоритм для побудови оптимального маршруту між двома точками.

2. У сфері телекомунікацій алгоритм Флойда-Уоршола використовується для оптимізації шляхів передачі даних. Він може бути застосований для визначення найкоротших шляхів передачі сигналів у комунікаційних мережах, що дозволяє забезпечити оптимальну швидкість і надійність передачі даних.

3. У фінансовій аналітиці алгоритм Флойда-Уоршола може бути використаний для аналізу фінансових ризиків і портфельів. Він дозволяє знайти найбільш ефективний шлях розподілу ресурсів або інвестицій між різними активами або ринками з урахуванням ризиків і потенційних виправданих доходів.[3]

Алгоритм Беллмана-Форда є ефективним способом знаходження найкоротших шляхів від однієї вихідної вершини до всіх інших вершин у зваженому орієнтованому графі. Він може обробляти графи з ребрами негативної ваги, що робить його універсальним інструментом для широкого спектра задач.

1. Хоча алгоритм Беллмана-Форда є трохи повільнішим за алгоритм Дейкстри для тієї ж задачі, він пропонує більш гнучкий підхід, який дозволяє працювати з графами, де деякі ребра можуть мати від'ємну вагу [3].

Приклади використання алгоритму:

2. Вантажні компанії можуть використовувати його для знаходження найкоротших шляхів доставки вантажів, враховуючи різні фактори, такі як відстань, час та витрати на паливо.

3. Алгоритм Беллмана-Форда-Уоршола може бути застосований для аналізу соціальних мереж, де вершини представляють користувачів, а ребра - зв'язки між ними. Він допомагає знайти найкоротші шляхи сполучення між користувачами та виявити ключові взаємозв'язки в мереж

4. Алгоритм Беллмана-Форда-Уоршала використовується в маршрутизаторах комп'ютерних мереж для знаходження найкоротших шляхів передачі пакетів даних між вузлами мережі. Він дозволяє оптимізувати шляхи передачі даних з урахуванням затримки, пропускнуої спроможності та інших факторів [4].

1.2 Застосування нейромереж для прогнозування найкоротших шляхів

Нейромережі можуть бути застосовані для прогнозування найкоротших шляхів, наприклад, у задачах маршрутизації мережі.

Одним з методів застосування нейромереж для прогнозування найкоротших шляхів є використання штучних нейронних мереж для визначення оптимального маршруту між двома точками у мережі.

Зазвичай для цього використовують нейронні мережі з прямим поширенням (feedforward neural networks), що використовуються для класифікації, регресії або передбачення. За допомогою таких мереж можна передбачити найкоротший шлях на основі різноманітних вхідних даних, таких як відстань, час, кількість пересадок, обмеження на швидкість тощо. [5]

Наприклад, у мережі можна зберегти інформацію про відстані між різними точками у мережі, а також про максимальну швидкість на кожному з ділянок. Тоді за допомогою нейронної мережі можна знайти найкоротший шлях між двома точками, використовуючи ці дані та додаткову інформацію, таку як вхідні та вихідні точки, обмеження на швидкість та інші обмеження.

Іншим методом застосування нейромереж для прогнозування найкоротших шляхів є використання рекурентних нейронних мереж (RNN), які можуть враховувати послідовність даних. За допомогою RNN можна передбачити найкоротший шлях на основі історії маршруту та попередніх даних про шляхи.

Загалом, застосування нейромереж для прогнозування найкоротших шляхів є потужним інструментом, який може бути використаний для вирішення багатьох різних задач маршрутизації [5].

Загалом, класичні алгоритми знаходження найкоротшого шляху і застосування нейромереж в цій області досліджень доповнюють один одного і мають широкий потенціал для вирішення складних задач маршрутизації та оптимізації в мережах та графах. Подальші дослідження в цій галузі можуть привести до нових інноваційних рішень та покращення різних сфер застосування, що вимагають ефективного вирішення проблеми знаходження найкоротшого шляху.

Список літератури

1. Алгоритм Дейкстри URL:
https://jetiq.vntu.edu.ua/fdb/896/ОКМ_методичні_вказівки_v001.pdf
2. Приклад використання алгоритму Дейкстри, URL:
<https://www.mathros.net.ua/rozvjazok-zadachi-pro-najkorotshyj-shljah-vykorystovujuchy-algorytm-bellmana-forda.html>

3. Алгоритм Флойда-Уоршола, URL: <https://ekmair.ukma.edu.ua/items/427eaa2a-7fc8-4804-a132-7f5a20db7189>
4. Алгоритм Беллмана-Форда URL: <https://www.youtube.com/watch?v=8-YoNjGTImU>
5. Використання нейромереж для знаходження найкоротшого шляху, URL: <https://disted.edu.vn.ua/media/doc/Алгоритм%20Дієкстри.pdf>

УДК 004.01

*Леценко В. О., здобувач 4 курсу
Зелінська О.В, к.т.н, доцент,
доцент кафедри інформаційних наук*

СЕРВЕРНА ЧАСТИНА ВЕБ-САЙТУ ОНЛАЙН-КНИГАРНІ

Донецький національний університет імені Василя Стуса

Зростаюча популярність онлайн-книгарень свідчить про зміну способу споживання книжок. Сучасні читачі все частіше обирають зручний доступ до електронних книг через Інтернет. Це зумовлено такими факторами, як мобільність, широкий вибір літератури, можливість здійснювати покупки та читати книжки безпосередньо зі своїх пристроїв. Онлайн-книгарні стають популярними майданчиками для здійснення книжкових покупок, обміну рекомендаціями та спілкування зі спільнотою книголюбів [1].

Розвиток технологій та зростання ролі Інтернету вимагають ефективних серверних рішень для онлайн-книгарень. Інтернет став неодмінною складовою нашого повсякденного життя, а розширення можливостей онлайн-середовища поширюється на всі сфери діяльності, включаючи книгарні. Забезпечення швидкого та стабільного доступу до книжкового контенту, безпеки та зручності для користувачів стає важливим завданням. Відповідно, розробка ефективних серверних рішень є ключовим аспектом для успішного функціонування онлайн-книгарень.

Розуміння потреб користувачів є фундаментальним для розробки ефективної серверної частини [2]. Онлайн-книгарні повинні аналізувати очікування та поведінку клієнтів, щоб забезпечити персоналізовані рекомендації, зручні опції пошуку, надійну обробку платежів та ефективне управління замовленнями. Задовольняючи ці потреби, онлайн-книгарні можуть створити користувальницько-орієнтований досвід, який сприяє лояльності клієнтів і росту бізнесу.

Розробка масштабованої і гнучкої архітектури серверної частини є важливою для роботи зі зростаючим трафіком користувачів та забезпечення надійності системи. Архітектура повинна включати кілька рівнів, таких як веб-сервери, сервери застосунків та сервери баз даних, для ефективної обробки різних завдань [3]. Шляхом використання технологій, таких як балансування навантаження та кешування, онлайн-книгарні можуть розподілити робоче навантаження і забезпечити швидку та стабільну роботу для користувачів [4].