

## ОСОБЛИВОСТІ ВИКОРИСТАННЯ ТЕОРІЇ ГРАФІВ ПРИ ПРОЕКТУВАННІ КАРТОГРАФІЧНИХ СЕРВІСІВ

*Анотація.* У статті проведено детальний аналіз особливостей впровадження елементів теорії графів у розвиток картографічних сервісів. Розглянуті проблеми, що виникають під час вибору алгоритму пошуку найкоротшого шляху у графі. Визначено можливості використання різних методів обходу графа. Досліджено особливості впровадження даних алгоритмів відповідно до можливостей, що надаються найбільш прогресивними лідерами веб-картографії.

*Ключові слова:* API, картографічний сервіс, граф

### 1 Вступ

В умовах епохи інформаційних технологій значне поширення отримала веб-картографія. У сьогоденні на базі всесвітньої павутини стрімко формується інтерактивна глобальна інфраструктура веб-картографії, яка нараховує, окрім професіоналів, мільйони пересічних користувачів по усій планеті. В таких умовах виникає потреба у створенні додатків для освоєння даної галузі, її популяризації та підлаштування відповідно до потреб кінцевого користувача. Тому постає проблема вибору алгоритму пошуку оптимального маршруту, що найкраще реалізовано з використанням елементів теорії графів. Задача визначення найкоротшого шляху графа може бути визначена для орієнтованого, неорієнтованого або комбінованого графів. Відповідно до даних критеріїв розглянемо найвідоміші алгоритми.

## 2 Основна частина

В останні часи теорія графів залучає все більше уваги спеціалістів різних областей знань. Поряд з традиційними застосуваннями її в таких науках, як фізика, електротехніка, хімія, вона проникла в такі науки, що раніше вважалися далекими від неї – лінгвістику, економіку, соціологію та інші. Особливо важливий зв'язок між теорією графів з теоретичною кібернетикою (особливо теоріями автоматів, кодування та іншими) [1].

На рис. 1 зображено доцільність використання графів для роботи з географічними даними. На даному зображенні утворено граф, вершинами якого є країни Південної Америки, а ребрами – їх зв'язки (у даному випадку – кордони) з іншими державами [2].

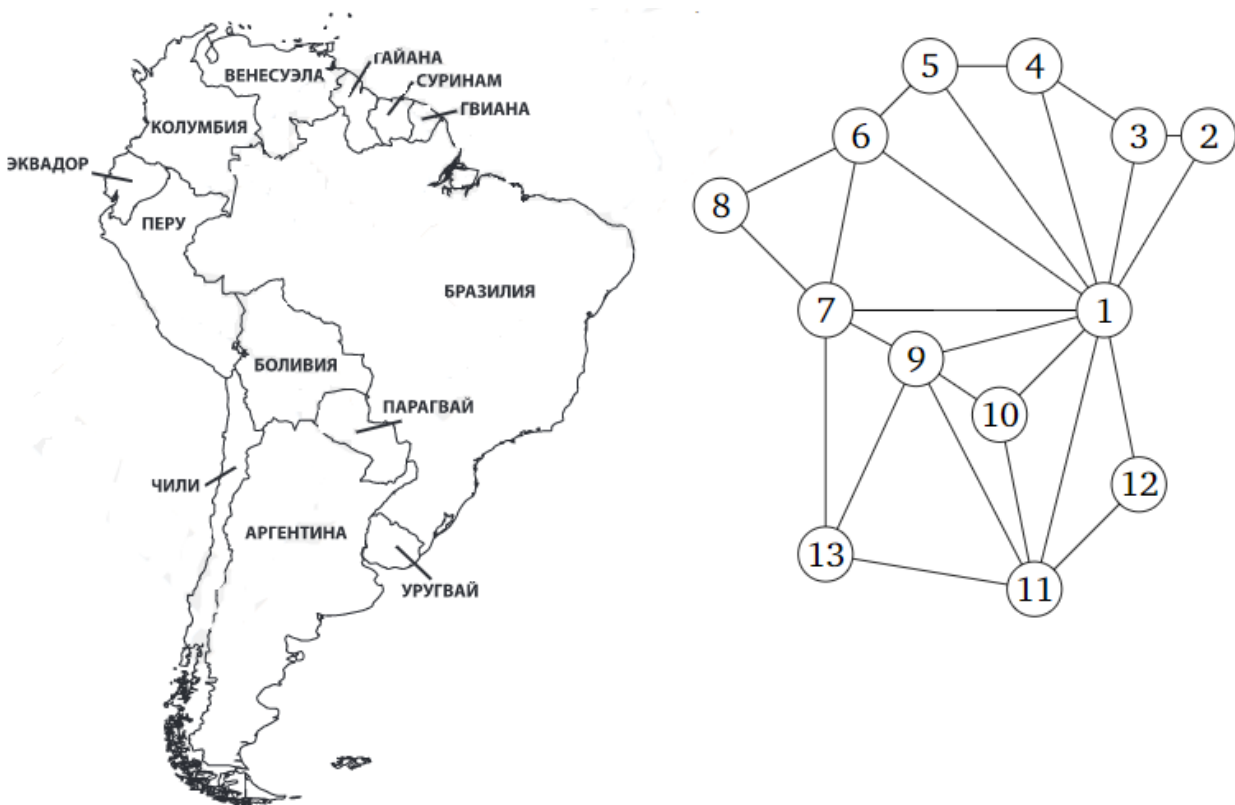


Рис. 1. Аналогія зовнішніх зв'язків країн зі структурою графу

Розглянемо Алгоритм Дейкстри. Він призначений для зваженого графа з невід’ємною вагою ребер.

Кожній вершині графу надається мітка, що є оцінкою мінімального шляху в неї із стартової вершини. Алгоритм є покроковим, тобто на кожному кроці він розглядає одну вершину та намагається зменшити мітки суміжних вершин. Робота алгоритму добігає до кінця у випадку відвідування кінцевої вершини. Від самого початку стартова мітка є рівною нулю, мітки інших вершин – нескінченності (нескінченність каже про те, що довжина шляху до вершини на даний час невідома. Усі вершини графу позначаються як невідвідані (рисунок 2).

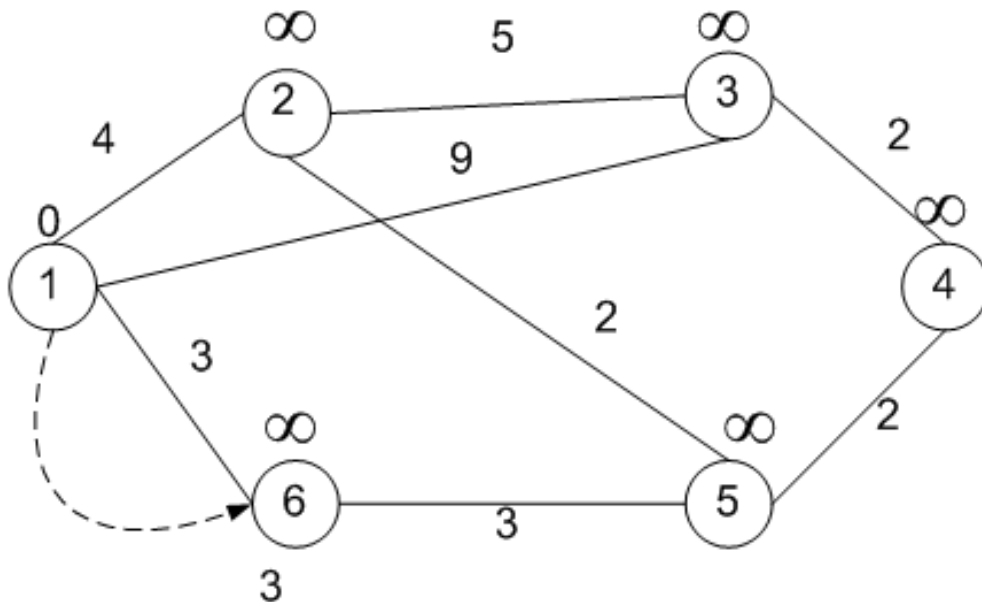


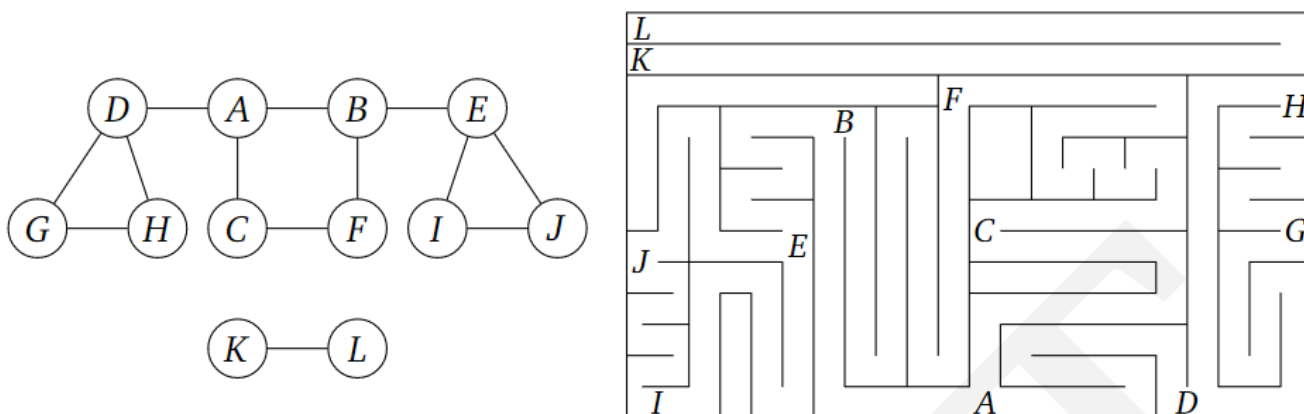
Рис. 2. Початковий етап алгоритму Дейкстри

Час виконання алгоритму залежить від способу зберігання множини невідвіданих вершин та способу оновлення міток. Нехай  $V$  – кількість вершин, а  $E$  – кількість ребер. В загальному випадку пошуку вершини з мінімальним значенням шляху розглядається вся множина вершин, а для зберігання їх використовується масив, тоді час роботи алгоритму складає  $O(V^2)$ . Для розріджених графів (таких, для котрих  $E$  менше за  $V^2$ ) невідвідвані вершини

можна зберігати у двійковій кучі, а в якості ключа використовувати значення найкоротшого шляху. Тоді час роботи складатиме  $O(V \cdot \log V + E \cdot \log V)$ .

Трохи інша картина виникає при використанні пошуку в ширину. Алгоритм призначений в основному для неорієнтованих графів, хоча існує варіація для орієнтовного графу.

Пошук в глибину (depth-first search) відповідає на питання: які вершини графа доступні із заданої вершини? Даний алгоритм найбільше підходить для задачі обходу лабіринту: маючи граф як список суміжності, у кожній вершині «бачимо» її сусідів – як у лабіринті, де ми бачимо сусідні приміщення (рис. 3), та можемо в них перейти. Але якщо ми будемо це робити без чіткого алгоритму, то можемо ходити колами, або зовсім не потрапити у пункт призначення.



*Рис. 3. Лабіринт та його граф*

Виконати обхід лабіринту можливо за допомогою клубка ниток (закріпивши нитку в початковій точці і носячи його за собою, ми можемо завжди повернутися) та крейди (щоб помічати місця, де ми вже були). Приблизно так працює алгоритм пошуку в глибину.

Головна відмінність від пошуку в ширину полягає у тому, що при пошуку в глибин у якості активної обирається та з відкритих вершин, яка була відвідана останньою.

Під час пошуку в ширину ідея полягає у тому, щоб відвідувати вершини у порядку їх віддаленості від деяких відпочатку обраних або вказаної початкової

вершини А. Інакше кажучи, спочатку відвідується сама вершина А, потім усі суміжні з нею, тобто ті, що знаходяться на відстані 1, далі вершини, що знаходяться на відстані 2, і так далі (рис. 4).

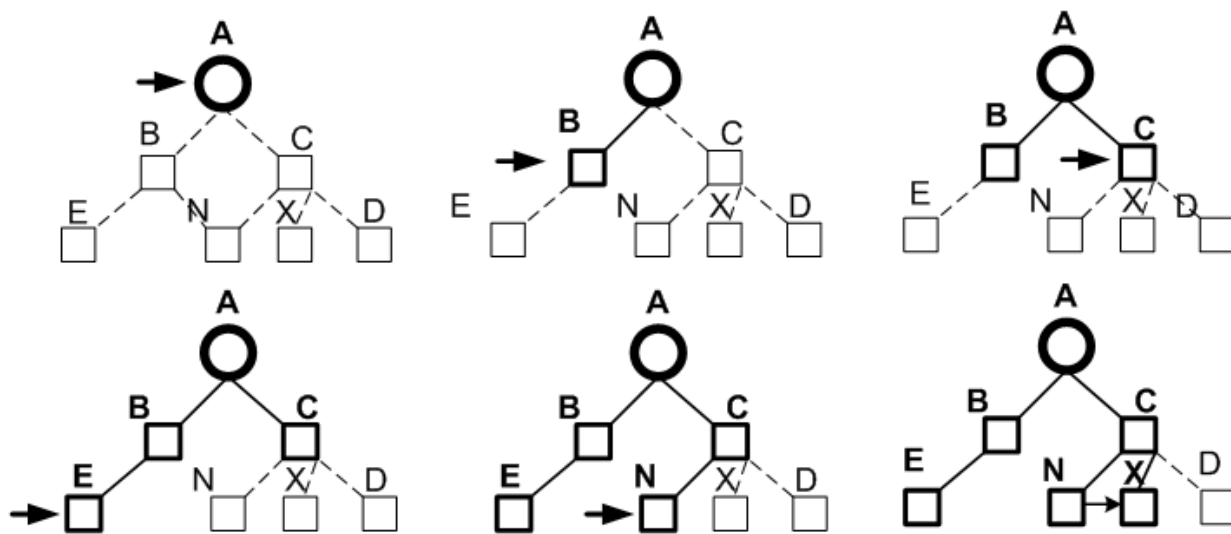


Рис. 4. Схема алгоритму пошуку в ширину

Більш детально виправданість використання даного алгоритму для картографічних сервісів, що орієнтовні на русі міського транспорту, бачимо у наступному прикладі. Нехай, маємо граф (рис. 5).

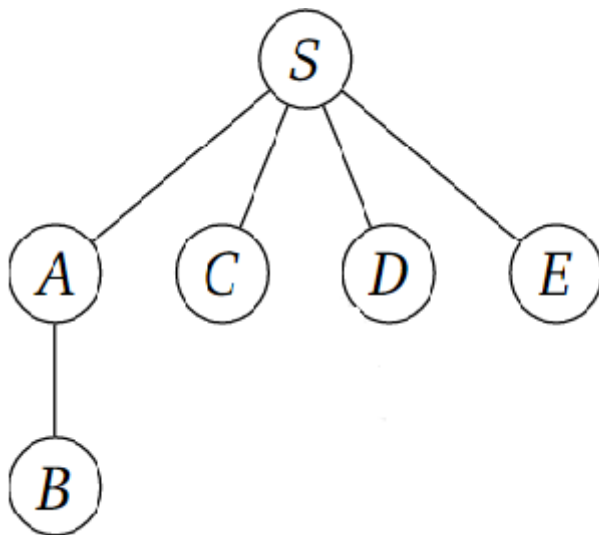


Рис. 5. Граф для обходу в ширину

Даний граф є деревом пошуку в ширину, а у таблиці 1 наведено порядок обходу вершин. Це дерево має лише ребра, при переході по яких віднаходяться нові вершини. На відміну від дерева пошуку в глибину, всі шляхи даного дерева з початком в S є найкоротшими. Тому воно називається деревом найкоротших шляхів [2].

*Таблиця 1*

*Порядок відвідування вершин графа*

Порядок відвідування	Вміст черги після обробки вершини
	[S]
S	[ACDE]
A	[CDEB]
C	[DEB]
D	[EB]
E	[B]
B	[]

Так, як API веб-карт відкриває доступ до програмування на основі мобільної розробки (переважно на мові Java) або через мову веб-програмування JavaScript, то найбільш виправданим з точки зору простоти реалізації та мобільності використовувати серверні мови програмування для взаємодії з відкритими ресурсами картографічних сервісів, інтерфейс взаємодії налагодивши через мову JavaScript.

### **3 Висновок**

Розглянуто основні алгоритми пошуку оптимального маршруту у графі та їх доцільність використання при розробці транспортних маршрутів. Вибір кожного з них буде зумовлено відповідно до поставленої задачі. При необхідності побудови маршруту, у якого шлях (ребра) матимуть вагу (тобто

граф буде зважений), впроваджено орієнтованість - доцільно використати алгоритм Дейкстри. В свою чергу, при нехтуванні відстанню між зупинками, опираючись на лише на їх кількість (у випадку, коли відстань приблизно однакова), оптимальним є метод пошуку в ширину.

## Посилання

1. *Теория графов. Ф.Харари /М.: Мир, 1973, 306с.*
2. *Алгоритмы. С. Дасгунта, Х. Панадимитриу, У. Вазирани; Пер. с англ. под ред. А. Шеня. М.: МЦНМО, 2014. - 320 с.*

**Authors:** Andriy Shugaylo, Somov Serhii

### **Features of application of the graph theory in case of design of cartographical services**

**Abstract.** In article the detail analysis of features of implementation of elements of the graph theory is carried out to development of cartographical services. The problems arising in case of a choice of a shortest path algorithm in the graph are considered. Possibilities of use of different methods of bypass of a graph are defined. Features of implementation of these algorithms according to opportunities which are given by the most progressive leaders of web cartography are probed.

**Keywords:** API, cartographical service, graph.

**Авторы:** Шугайло Андрей Олегович, Сомов Сергей Викторович

### **Особенности использования теории графов при проектировании картографических сервисов**

**Аннотация.** В статье проведен детальный анализ особенностей внедрения элементов теории графов в развитие картографических сервисов. Рассмотрены проблемы, возникающие при выборе алгоритма поиска кратчайшего пути в графе. Определены возможности использования различных методов обхода графа. Исследованы особенности внедрения данных алгоритмов в соответствии с возможностями, которые предоставляются наиболее прогрессивными лидерами веб-картографии.

**Ключевые слова:** API, картографические сервисы, граф.