УДК 512.62

АНАЛІЗ ЗАСТОСУВАННЯ МЕТОДІВ ТЕОРІЇ ГРАФІВ У ДИСЦИПЛІНІ «ДИСКРЕТНА МАТЕМАТИКА»

Юник Д., студент групи МП-31-18
ХНАДУ

Анотація. Серед дисциплін і методів дискретної математики теорія графів і особливо алгоритми на графах знаходять найбільш широке застосування в програмуванні. Як правило, є безліч різних способів рішень задач за допомогою графів. Наприклад, це можуть бути завдання, пов'язані з пошуком шляху між початковою і кінцевою точками або пошуком максимальної кількості рідини, яке може пройти через трубопровід, якщо відома максимальна пропускна здатність кожної ділянки трубопроводу.

Ключові слова: дискретна математика, граф, шлях на графі, орієнтований граф, маршрути графа, зв'язність графа.

Графом називається множина вузлів і зв'язків між ними. Кожен вузол називається вершиною, а кожен зв'язок – ребром. Кожне ребро з'єднує дві вершини.

Однак теорія графів виникла задовго до появи програмування. Її появу пов'язують зі старовинною математичною задачею про сім кенігсберзьких мостів, яка стала класичною. Колишній Кенігсберг (нині Калінінград) розташований на річці Прегель. У межах міста річка омиває два острови. З берегів на острови були перекинуті мости. Старі мости не збереглися, але залишилася карта міста, де вони зображені. Кенігсбергці пропонували приїжджим наступне завдання: пройти по всіх мостах і повернутися в початковий пункт, причому на кожному мосту слід було побувати тільки один раз. Втім, довести або спростувати можливість існування такого маршруту ніхто не міг. У 1736 році задача про сім мостів зацікавила видатного математика, члена Петербурзької академії наук Леонарда Ейлера, про що він написав у листі італійському математику та інженеру Маріоні від 13 березня 1736 року. У цьому листі Ейлер пише про те, що він зміг знайти правило, користуючись яким, легко визначити, чи можна пройти по всіх мостах, не проходячи двічі ні по одному з них. Відповідь була «не можна».

Розбір Ейлера

По-перше, Ейлер осягнув, що вибір маршруту всередині кожної з ділянок суходолу (островів, або берегів ріки) не має значення. Важлива лише послідовність перетину мостів. Це дозволило йому переформулювати задачу в абстрактних термінах (які лягли в основу теорії графів), виключивши усі ознаки окрім списку ділянок суходолу і мостів, що сполучають їх. В сучасних термінах, він кожну з ділянок суходолу замінив на абстрактну «вершину», а кожен міст на абстрактне «ребро», яке слугувало лише для відображення факту сполучення пари вершин (ділянок суходолу) цим мостом. Отримана математична структура називається графом.

Через те, що важлива лише інформація про зв’язки, форма в якій граф зображений на малюнку не має значення, якщо при цьому не змінюється сам граф. Тільки існування (або відсутність) ребра між кожною парою вершин має значення. Наприклад, не має значення чи ребра намальовані як прямі або криві, або праворуч чи ліворуч від іншого зображений вузол.

Наступним спостереженням Ейлера було те, що (окрім кінцевих вершин прогулянки), коли хтось потрапляє до вершини через міст, обов’язково її покидає через міст. Інакше кажучи, впродовж кожного маршруту в графі, кількість входів в некінцеві вершини дорівнює кількості виходів з них. Тепер, якщо кожний міст пройдено рівно один раз, вірно наступне, для кожної ділянки суходолу (окрім початкової і кінцевої), кількість мостів до цієї ділянки парна (половина з них буде пройдена за напрямком до ділянки, а половина з ділянки). Однак всі чотири ділянки суходолу в початковій задачі мають непарну кількість мостів (одна 5, а інші по 3). Через те, що лише дві ділянки можуть слугувати як початкова і кінцева точки ймовірного маршруту, задача пройтися усіма мостами рівно по одному разу призводить до протиріччя.

Сучасною мовою, Ейлер показав, що можливість пройти через граф, пройшовши кожне ребро рівно один раз, залежить від степенів вершин. Степінь вершини це кількість ребер, що торкаються її. Аргументи Ейлера показали, що необхідною умовою прогулянки бажаного виду через граф є зв'язність графа і відсутність або наявність рівно двох вершин непарного степеня. Ця умова виявилась і достатньою, що стверджував Ейлер і пізніше довів Карл Гьєхолзер. Такий шлях називається ейлерів шлях.

Далі, якщо присутні дві вершини непарного степеня, тоді Ейлерів шлях почнеться з однієї з них і закінчиться в іншій. Через наявність чотирьох вершин непарного степеня, історична задача не має розв'язку.

Іншим формулюванням задачі є запит на шлях, який проходить усіма ребрами і початкова та кінцева точки якого збігаються. Такий шлях називаєть ейлерів цикл. Такий шлях існує тоді і тільки тоді, коли граф зв'язний і не містить вершин непарного степеня. Всі ейлерові цикли є ейлеровими шляхами, але не навпаки. Робота Ейлера була представлена Петербурзькій Академії 26 серпня 1735 року.

В історії математики, Ейлерів розв'язок задачі Кеніґсберзьких мостів вважається першою теоремою теорії графів (сума степенів всіх вершин завжди парна), задача розглядається як відгалуження комбінаторики. Комбінаторні задачі інших типів розглядались з античних часів.

Простим графом G (V, E) називається сукупність двох множин – непорожньої множини V і множини E невпорядкованих пар різних елементів множини V. Множина V називається множиною вершин, множина E називається множиною ребер. p – число вершин, q – число ребер. Графи можна представити різними способами (що буде розглянуто далі), але зазвичай граф зображують діаграмою.

Якщо елементами множини E є впорядковані пари (тобто пари, в яких фіксований порядок елементів), то граф називається орієнтованим (або орграфом). У цьому випадку елементи множини V називаються вузлами, а елементи множини E – дугами. Першу вершину впорядкованої пари називають початком дуги, другу – кінцем.

Нехай v_1, v_2-вершини, $e = v_1v_2$ - ребро, що з'єднує їх. Тоді вершина v_1 і ребро e інцидентні, вершина v_2 і ребро e також інцидентні. Два ребра, інцидентні одній вершині, називаються суміжними, дві вершини, інцидентні одному ребру, також називаються суміжними.

Множина вершин, суміжних з вершиною v, називається множиною суміжності вершини v і позначається Г (v) = {u: vu ∈ E}. Якщо A ⊂ V - множина вершин, то Г(A) - множина всіх вершин, суміжних з вершинами з A: $Г(A) = \cup_{v\in a} Г(v)$.

Кількість ребер, інцидентних вершині v, називається ступенем(або валентністю) вершини v і позначається d(v). Якщо ступінь вершини дорівнює 0, то вершина називається ізольованою. Якщо ступінь вершини дорівнює 1, то вершина називається висячою.

Маршрутом в графі називається послідовність вершин і ребер виду $v_0e_1v_1e_2... v_{k-1}e_kv_k$, в якій $e_i = v_{i-1}v_i$. Вершина v_0 називається початковою, а v_k – кінцевою вершиною маршруту. Для графа без кратних ребер досить вказати тільки послідовність вершин. Якщо всі ребра в маршруті різні, то маршрут називається ланцюгом. Якщо всі вершини (а значить і ребра) в маршруті різні, то маршрут називається простим ланцюгом.

Якщо $v_0=v_k$, маршрут називається замкнутим. Якщо всі ребра в замкнутому маршруті різні, то він називається циклом. Якщо всі вершини в замкнутому маршруті, крім першої і останньої, різні, то він називається простим циклом. В орграфі ланцюг називається шляхом, а цикл - контуром.

Граф $G_0(V_0,E_0)$ називається підграфом графу G(V,E) (позначається $G_0 \subseteq G$), якщо $V_0 \subseteq V$ та $E_0 \subseteq E$. Якщо $V_0 = V$, то підграф $G_0(V_0,E_0)$ називається остовним підграфом графа G(V,E).

Якщо $V_0 \neq V$, $E_0 \neq E$ то підграф G_0V_0, E_0 називається власним підграфом графа $G(V,E)$. Підграф $G_0(V_0,E_0)$ називається правильним підграфом графа $G(V,E)$ якщо він містить всі можливі ребра вихідного графа: $\forall u$, $v \in V_0$: $uv \in E \Rightarrow uv \in E_0$.

Дві вершини в графі зв’язні, якщо існує ланцюг, що з’єднує їх. Граф називається зв’язним, якщо будь-які дві вершини в ньому зв’язні. Компонентою зв’язності графа $G(V,E)$ називається його правильний зв’язний підграф, що не є власним підграфом ніякого іншого зв’язного підграфа графа $G(V,E)$.

Граф G (V,E) називається тривіальним, якщо він складається з однієї вершини: р = 1, Е = ∅. Граф G (V,E) називається порожнім, якщо Е = ∅. Граф G (V,E) називається повним, якщо в ньому будь-які дві вершини суміжні, тобто $\forall u,\ v \in V$: $uv \in E$ (позначення K_p). Якщо ступені всіх вершин рівні k, то граф називається регулярним ступеня k. Граф $G(V,E)$ називається дводольним, якщо множину V можна розбити на дві підмножини V_1 і V_2, що не перетинаються таким чином, що будь-яке ребро з Е з'єднує вершину з V_1 з вершиною з V_2 . Множини V_1 і V_2 називаються частками дводольного графа. Якщо дводольний граф містить всі можливі ребра, тобто $\forall v_1 \in V_1,\ v_2 \in V_2$: $v_1 v_2 \in E$, то він називається повним дводольним графом і позначається $K_{m,n}$, де $m=|V_1|$, $n=|V_2|$. Зваженим графом $G(V,E)$ називається граф, кожному ребру якого зпівставлено деяке число, що називається вагою.

Операції над графами. Об'єднання графів $G_1(V_1,E_1)$ $G_2(V_2,E_2)$ $G(V,E)=G_1(V_1,E_1) \cup G_2(V_2,E_2)$. З'єднання графів $G_1(V_1,E_1)$ $G_2(V_2,E_2)$ $G(V,E)=G_1(V_1,E_1) + G_2(V_2,E_2)$. Доповнення графу $G_1(V_1,E_1)$ $G_2(V_2,E_2)=$ $G_1(V_1,E_1)$. Стягування правильного підграфа у вершину.

Матриця суміжності графа - це квадратна матриця, в якій кожен елемент приймає одне з двох значень: 0 або 1. Число рядків матриці суміжності дорівнює числу стовпців і відповідає кількості вершин графа.

- 0 - відповідає відсутності ребра,
- 1- відповідає наявності ребра.

Коли з однієї вершини в іншу прохід вільний (є ребро), в комірку заноситься 1, інакше – 0. Всі елементи на головній діагоналі дорівнюють 0 якщо граф не має петель.

Матриця інцидентності (инциденції) графа – це матриця, кількість рядків в якій відповідає числу вершин, а кількість стовпців – числу ребер. У ній вказуються зв'язки між інцидентними елементами графа (ребро (дуга) і вершина).

У неорієнтованому графі якщо вершина інцидентна ребру то відповідний елемент дорівнює 1, в іншому випадку елемент дорівнює 0. В орієнтованому графі якщо ребро виходить з вершини, то відповідний елемент дорівнює 1, якщо ребро входить в вершину, то відповідний елемент дорівнює -1, якщо ребро відсутнє, то елемент дорівнює 0. Матриця інцидентності для свого представлення вимагає нумерації ребер, що не завжди зручно.

Список суміжності (інцидентності). Якщо кількість ребер графа в порівнянні з кількістю вершин невелика, то значення більшості елементів матриці суміжності дорівнюватимуть 0. При цьому використання даного методу недоцільно. Для подібних графів є більш оптимальні способи їх представлення.

По відношенню до пам'яті списки суміжності менш вимогливі, ніж матриці суміжності. Такий список можна представити у вигляді таблиці, стовпців в якій – 2, а рядків – не більше, ніж вершин в графі. У кожному рядку в першому стовпці вказана вершина виходу, а в другому стовпці – список вершин, в які входять ребра з поточної вершини.

Переваги списку суміжності:

- Раціональне використання пам'яті.
- Дозволяє швидко перебирати сусідів вершини.
- Дозволяє перевіряти наявність ребра і видаляти його.

Недоліки списку суміжності:

- При роботі з насиченими графами (з великою кількістю ребер) швидкості може не вистачати.
- Немає швидкого способу перевірити, чи існує ребро між двома вершинами.
- Кількість вершин графа має бути відома заздалегідь.
- Для зважених графів доводиться зберігати список, елементи якого повинні містити два значущих поля, що ускладнює код:
  - номер вершини, з якої з'єднується поточна;
  - вага ребра.

Список ребер.

У списку ребер в кожному рядку записуються дві суміжні вершини і вага ребра, що їх з'єднує (для зваженого графа). Кількість рядків у списку ребер завжди має дорівнювати величині, що виходить в результаті складання орієнтованих ребер з подвоєною кількістю неорієнтованих ребер.

Який спосіб подання графа краще? Відповідь залежить від відношення між числом вершин і числом ребер. Число ребер може бути досить малим (такого ж порядку, як і кількість вершин) або досить великим (якщо граф є повним). Графи з великим числом ребер називають щільними, з малим – розрідженими. Щільні графи зручніше зберігати у вигляді матриці суміжності, розріджені – у вигляді списку суміжності.

Алгоритми обходу графів.

Основними алгоритмами обходу графів є

- Пошук в ширину
- Пошук в глибину

Пошук в ширину має на увазі порівневе дослідження графа:

- спочатку відвідується корінь-довільно обраний вузол,
- потім-всі нащадки даного вузла,
- після цього відвідуються нащадки нащадків і т. д.

Вершини проглядаються в порядку зростання їх відстані від кореня. Алгоритм припиняє свою роботу після обходу всіх вершин графа, або в разі виконання необхідної умови (наприклад, знайти найкоротший шлях з вершини 1 в вершину 6).

Кожна вершина може перебувати в одному з 3 станів:

- 0 – Помаранчевий – невиявлена вершина;
- 1 – Зелений – виявлена, але не відвідана вершина;
- 2 – Сірий – оброблена вершина.

Фіолетовий – розглянута вершина.

Застосування алгоритму пошуку в ширину.

- Пошук найкоротшого шляху в незваженому графі (орієнтованому або неорієнтованому).
- Пошук компонент зв’язності.
- Знаходження рішення будь-якої задачі (гри) з найменшим числом ходів.
- Знайти всі ребра, що лежать на будь-якому найкоротшому шляху між заданою парою вершин.
- Знайти всі вершини, що лежать на будь-якому найкоротшому шляху між заданою парою вершин.

Алгоритм пошуку в ширину працює як на орієнтованих, так і на неорієнтованих графах.

Для реалізації алгоритму зручно використовувати чергу.

Пошук в глибину - це алгоритм обходу вершин графа.

Пошук в ширину проводиться симетрично (вершини графа проглядалися за рівнями). Пошук в глибину передбачає просування вглиб до тих пір, поки це можливо. Неможливість просування означає, що наступним кроком буде перехід на останній, що має кілька варіантів руху (один з яких досліджений повністю), раніше відвіданий вузол (вершина).

Відсутність останнього свідчить про одну з двох можливих ситуацій:

- всі вершини графа вже переглянуті,
- переглянуті вершини доступні з вершини, взятої в якості початкової, але не всі (незв'язні і орієнтовані графи допускають останній варіант).

Кожна вершина може перебувати в одному з 3 станів:

- 0 – Помаранчевий – невиявлена вершина;
- 1 – Зелений – виявлена, але не відвідана вершина;
- 2 – Сірий – оброблена вершина;

Фіолетовий – розглянута вершина.

- Застосування алгоритму пошуку в глибину
- Пошук будь-якого шляху в графі.
- Пошук лексикографічно першого шляху в графі.
- Перевірка, чи є одна вершина дерева предком інший.
- Пошук найменшого спільного предка.
- Топологічне сортування.
- Пошук компонент зв'язності.

Алгоритм пошуку в глибину працює як на орієнтованих, так і на неорієнтованих графах. Застосовність алгоритму залежить від конкретного завдання.Для реалізації алгоритму зручно використовувати стек або рекурсію.

Таким чином, від старовинної задачі про мости виник цілий підрозділ дискретної математики – теорію графів. Вона застосовується при вирішенні найрізноманітніших задач. З плином часу графи досліджувалися та класифікувалися, але найбільшого розвитку теорія графів набула з появою обчислювальної техніки. Зараз є декілька видів представлення графів у пам'яті комп'ютера, кожен з яких має свої переваги та недоліки.

Література

1. Алгебра и начала анализа. / Часть 1. Под общ. ред. Г.Н. Яковлева. – М.: Наука, 1981. – 251 с.
2. Бардачов Ю.М. Дискретна математика / Ю.М. Бардачов, Н.А. Соколова, В.Є. Ходаков. – Київ: Вища шк., 2007. – 383 с.
3. Берж К. Теория графов и ее применение / К. Берж М.: ИЛ, 1962. – 224 с.
4. Бондаренко М.Ф., Білоус Н.В. Руткас А.Г. Комп'ютерна дискретна математика. – Харків: Компанія «СМІТ», 2004. – 480 с.
5. Дискретная математика и математические вопросы кибернетики / Под общ. ред. С.В. Яблонского, О.Б. Лупанова. Т.1. – М.: Наука, 1974. – 312 с.
6. Донской В.И. Дискретная математика / В.И. Донской. – Симферополь: Сонат, 2000. – 360 с.
7. Зыков А.А. Основы теории графов / А.А. Зыков. – М.: Наука, 1987. – 380 с.
8. Колодяжный В.М. Введение в дискретную математику: учебн. пос. / В.М. Колодяжный, И.Б. Сироджа. – Х.: ХАИ, 1999. – Часть 1. – 160 с.; Часть 2. – 204 с.