Лекція 09.11.2020 р.
Основні поняття теорії графів

Рівняння, складені за законами Кірхгофа, визначаються тільки схемою з'єднань гілок , тобто геометричною структурою в ланцюзі. Вони не залежать від виду й характеристик елементів. Тому при складанні рівнянь Кірхгофа зручно відволіктися від виду й характеристик гілок, й замінити їх лініями, котрі з'єднуються вузлами. В результаті ми отримаємо граф електричного ланцюга.

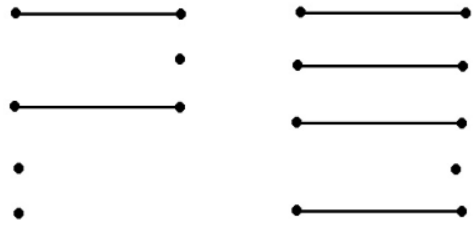
Геометричні властивості ланцюгів вивчає розділ теорії кіл, що називається топологією. Це одне з направлень математики, котре називається теорією графів. Детальний розбір питань топології електричних кіл не являється цілю лекції. Ми лише розглянемо основні поняття, котрі стосуються топології електричних ланцюгів.

Нескладний математичний апарат, який називають теорією графів, дає можливість ефективно розв'язувати задачі по аналізу розміщення служб, точок відеонагляду, постачання РЕА, електричних ліній зв'язку, формування мереж зв'язку, транспортні задачі, деякі задачі економіки та ін. Геометрична наочність і простота основних понять та великий ступінь загальності, що дає можливість різноманітні задачі розглядати на одній графовій моделі, обумовлюють широту практичних застосувань теорії графів.

Теорія графів — це розділ якісної геометрії. Якісна геометрія оперує безрозмірними величинами. Тут не використовуються ні поняття кута і одиниць його виміру (градусів), ні довжини ліній (метрів, сантиметрів). Головне в якісній геометрії — наявність просторових елементів — точок, ліній, поверхонь, об'ємів і відношень (зв'язків) між ними. Основним поняттям теорії графів є поняття графа.

Граф G = <V, E> – це абстрактний комбінаторний об'єкт, що складається з двох множин. V – довільна множина вершин, або об'єктів. Множина може бути нескінченною. E – множин ребер, або пар об'єктів з V.

Приклад 1. Розглянемо множину V студентів в аудиторії. Потужність множини |V| = 21. Якщо студенти сидять по двоє за партою і студентів позначимо точками, то студентів за однією партою можна позначити ребром. Маємо зображення, що показано на рис. 2.Із зображення зрозуміло , що парти стоять в два ряди, чотирнадцять студентів сидять по двоє за партою, п'ять студентів сидять по одному.

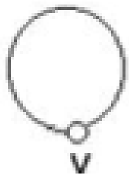


Розміщення студентів за партами

Ребро, яке з'єднує дві вершини графа, є інцидентним цим вершинам, а самі вершини — інцидентні ребру.

Дві вершини графа іменуються суміжними; якщо вони визначають ребро графа. Два ребра іменуються суміжними, якщо вони мають спільну вершину. Вершина, інцидентна тільки одному ребру, називається висячою, а не інцидентна жодному ребру — ізольованою. Граф, утворений тільки ізольованими вершинами — це нуль-граф.

Якщо кінці ребра інцидентні одній і тій же вершині, то таке ребро називається петлею. На рис. 3.15 зображене ребро-петля, яке виходить з вершини 7 і входить у неї.



Інколи пари вершин зєднуються не одним, а декількома ребрами – такі графи називають мультиграфами.

Якщо порядок кінців ребер графа не береться до уваги, то такий граф називається неорієнтованим (рис. 3.16, а). У протилежному випадку, тобто коли пари вершин упорядковані, граф називається орієнтованим (орограф).

Орієнтовані ребра — це дуги графа (рис. 3.16, б). Змішаним називається такий граф, в якому частина ребер орієнтована, а частина неорієнтована (рис. 3.16, б).

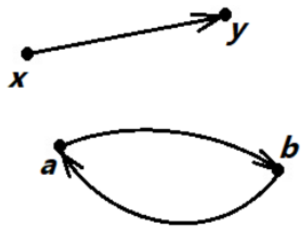


Рисунок 4 – Орієнтовані графи

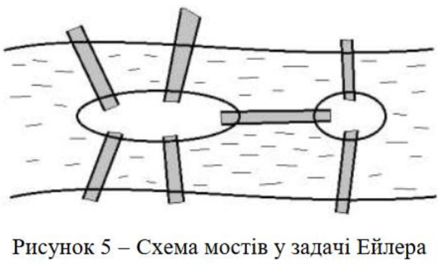


Рисунок 5 – Схема мостів у задачі Ейлера

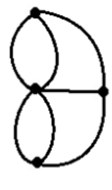


Рисунок 6 – Мультиграф до задачі Ейлера

Приклад. Розглянемо структуру певного Web-сайту де вершинами будуть сайти,а ребрами – гіперлінки. У цьому разі зображення графа може мати вигляд, наведений на рис. 7. Зліва на рис. 7 показано гіперлінк сторінки на саму себе. Ребро такого типу називається петля. Як видно з рис. 7, петлі можуть бути кратні. Граф, що містить лише петлі (x, x) називається псевдографом.

Рисунок 7 – Граф Web-сайту

Графи бувають скінченими, якщо множина їхніх вершин X = {x1, x2 , ...,x*n*} є скінченою, і нескінченними, якщо ця множина змінюється від одиниці до нескінченності.

Якщо граф має *n* вершин (*n* > 1) і кожна пара вершин з'єднана ребром, він називається повним. У протилежному випадку маємо неповний граф.

Існують поняття доповнення до графа, часткового графа і під-графа.

Доповнення до графа — це такий граф, ребра якого разом з цим графом утворюють повний граф (позначається через G).

Частковим називається граф, ребра якого є підмножиною цього графа і множина вершин якого збігається з множиною вершин цього графа б).

Підграфом називається граф, вершинами якого є підмножина вершин певного графа, а ребрами — усі ребра, обидва кінці яких інцидентні цим вершинам (рис. 3.17 в). Так, граф на рис. 3.17 (б) є підграфом графа на рис. 3.17 (а), а цей останній є його надграфом.

Підграф – частина графа, отримана шляхом видалення деяких гілок вихідного графа. Контур – замкнений шлях, що проходить через ряд гілок й вузлів.

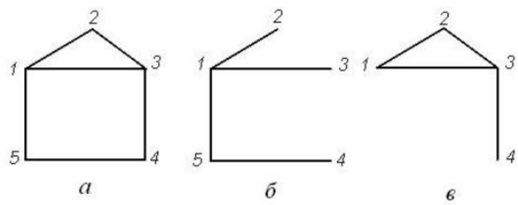


Рисунок 3.16 - Граф (а), його частковий граф (б), підграф (в)

Коли задається або відшукується певна послідовність ребер (дуг), тоді йдеться про маршрути — ланцюги і шляхи відповідно на неорієнтованому і орієнтованому графах, а також про цикли і контури теж відповідно на цих графах.

Ланцюг — така послідовність ребер неорієнтованого графа, в якій кінець одного ребра є початком іншого, а початкова і кінцева вершини не збігаються.

Шлях графа – неперервна послідовність гілок, з’єднаних парою обраних вузлів.

Маршрут в графі – це спосіб пройти в графові по ребрах – від вершини, до вершини. Тобто, маршрут – це послідовність v1e1v2e2...vnenvn+1з вершини v1 до вершини vn+1. Вершини vi до вершини vi+1 це кінці ребра ei. Можна записати ei=(vi,vi+1). Однак це не означає, що в маршруті всі ребра, або всі вершини, різні.

Якщо в маршруті всі ребра – різні і при цьому маршрут не замкнений, то маршрут носить назву – ланцюг. Наприклад, на рис. 8 v1e1v2e2v3e3v4e4v2

– ланцюг. Якщо в ланцюгу, або в циклі вершини не повторюються (крім першої, яка є останньою в циклі) то такий ланцюг, або цикл називається простим.

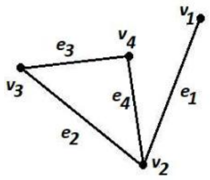


Рисунок 8 – Маршрут у простому графі

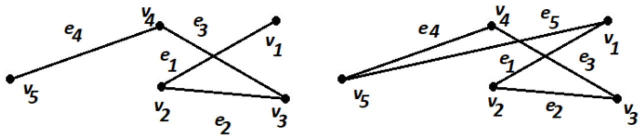


Рисунок 9 – Простий ланцюг (зліва) і простий цикл

Аналогічно визначається шлях на орграфі. Перша у ланцюгу чи шляху вершина називається початковою, остання — кінцевою, всі інші — проміжними.

Ланцюги чи шляхи можуть бути простими і складними, елементарними і неелементарними. Прості вони в тому випадку, коли усі ребра (дуги) різні. В протилежному випадку — це складні ланцюги чи шляхи.

Якщо у ланцюгу чи шляху жодна вершина не повторюється двічі, то він називається елементарним, і навпаки, ланцюг, який два або більше разів проходить через певну вершину, називається неелементарним.

Цикл — це ланцюг, який починається і закінчується однією і тією ж вершиною неорієнтованого графа. Аналогічно визначається контур орграфа.

Цикли і контури також бувають прості і складні, елементарні і неелементарні.

Це залежить від виду ланцюгів, які їх утворюють. На рис. 8, (а) маршрут 1-2, 2-6, 6-4, 4-5 є ланцюгом, а маршрут 1-2, 2-6, 6-4, 4-5, 5-1 — циклом. На рис. 3.16, (б) маршрут 1-2, 2-6, 6-4, 4-5 є шляхом. Цей же шлях плюс дуга 5-1 утворюють контур.

Виділяються також незалежні і залежні цикли чи контури. На рис. 3.16, а цикли 1-2, 2-4, 4-5, 5-1 та 2-6, 6-4, 4-2 незалежні, а цикл 1-2, 2-6, 6-4, 4-5, 5-1 залежний.

Довжина найкоротшого ланцюга між певними двома вершинами графа називається відстанню між ними. Якщо для кожної вершини підрахувати відстань до найбільш віддаленої від неї вершини, то найменше з цих чисел називатиметься радіусом, а найбільше — діаметром графа.

Дерево графа – підграф, котрий має в собі всі вузли графа, но не має контурів.

Гілки зв’язку (хорди) – гілки вихідного графа, котрі не ввійшли в дерево.

Приєднані кожною гілкою зв’язку до дерева дає контур, котрий називається головним. Кожна гілка зв’язку входить тільки в один контур. Тому рівняння по другому закону Кірхгофа для головних контурів будуть незалежними.

Переріз – система гілок , видалення яких розбиває граф на дві не зв’язні частини. Переріз розглядається як узагальнення поняття – вузол.

Поняття ланцюга чи шляху вихідне для розуміння фундаментального поняття зв'язаності графа.

Дві вершини графа називаються зв'язаними, якщо існує ланцюг чи шлях з кінцями в цих вершинах. Тоді неорієнтований граф вважається зв'язаним, якщо будь-яка пара його вершин зв'язана.

Для орієнтованих графів існує поняття дуже і малозв'язаних графів. Дуже зв'язаний — це орограф, в якому для будь-якої пари вершин існує шлях, що з'єднує їх. У протилежному випадку орограф малозв'язаний.

Графом-деревом називається довільний зв'язаний граф, який не має циклів (див. рис. 3.16, б). Ребра такого графа іменуються вішками, а ребра графа — доповнення до цього графа-дерева — хордами.

Існує 3 способи завдання графа:

1) графічний;

2) аналітичний

3) матричний.

1. Явний спосіб. Граф розглядається як дві множини G=<V, E> де V – множина вершин V={a, b, c, d} і E – множина ребер E={ (a, b), (a, c), (b,d), (b, c)}.

2. Графічний спосіб. Подання графа у вигляді зображення є поширеним, оскільки це зручний і наочний спосіб. Для графа G, описаного в способі 1, маємо зображення на рис. 12 зліва(1), або справа (2). Будь-яке з цих двох зображень графа G буде вірним і,навіть, можна сказати, що існує нескінченна множина таких зображень одного й того ж графа.

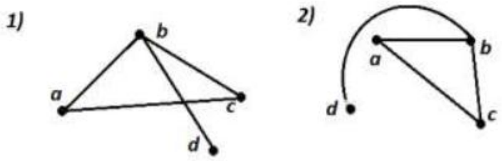
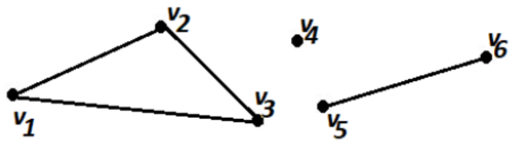


Рисунок 12 – Два зображення графа G

3. Матриця суміжності. Граф подається у вигляді матриці суміжності AG. Спосіб використовується при роботі комп'ютера з графами. Припустимо граф G має певну кількість вершин n, тобто - потужність множини вершин графа |V| = *n*. Тоді можна записати матрицю AG, розміром *n*x*n*, яка містить множину елементів AG= $(a_{i,j})_{nxn}$. Тут індекси i та j змінюються від 1 до n: $i, j = \overline{1, n}$, а значення $a_{i,j}$ приймає значення 1, коли між відповідними вершинами v_i та v_j існує ребро, або приймає значення нуль, якщо такого ребра не існує. Тобто,

$$a_{i,j} = \left[\left(v_i, v_j \right) \in E \right] = \begin{cases} 1, \left(v_i, v_j \right) \in E \\ 0, \left(v_i, v_j \right) \notin E \end{cases}$$

Як приклад, запишемо матрицю суміжності для графа G, показаного на рис. 10. Граф має шість вершин, тому n=6 і квадратна матриця має розмір 6x6.



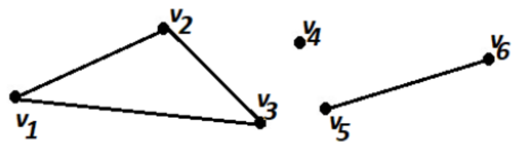
A_G =

	v1	v2	v3	v4	v5	v6
v1	0	1	1	0	0	0
v2	1	0	1	0	0	0
v3	1	1	0	0	0	0
v4	0	0	0	0	0	0
v5	0	0	0	0	0	1
v6	0	0	0	0	1	0

З означення матриці суміжності (2) витікає наступне: для не орієнтованого графа матриця суміжності завжди симетрична відносно головної діагоналі (відношення між вершинами є симетричними, користуючись апаратом теорії відношень); для не орієнтованого графа матриця відображає відношення сусідства вершин графа.

Матриця інцидентності. Матриця BG зв'язує вершини та ребра в деякому графі G. Як і попередній, спосіб використовується при роботі з графами на комп'ютері BG= $(b_{i,j})_{nx|E|}$ де $i, j = \overline{1, n}$, $b_{i,j}$ приймає значення 1, якщо вершина v_i є кінцем ребра e_j та навпаки – приймає значення нуль, якщо умова не виконується. У якості прикладу представимо граф G, поданий на рис. 10, матрицею інцидентності. Позначимо ребра графа наступним чином e1=(v1, v2), e2=(v2, v3), e3=(v1, v3), e4=(v5, v6). Маємо наступну матрицю інцидентності

	e1	e2	e3	e4
v1	1	0	1	0
v2	1	1	0	0
v3	0	1	1	0
v4	0	0	0	0
v5	0	0	0	1
v6	0	0	0	1



Для простого графа маємо дві важливі властивості матриці інцидентності. Перша полягає в наступному – кожен стовпчик матриці інцидентності містить рівно дві одиниці. Друга властивість матриці інцидентності – сума елементів кожної зі строк матриці дає значення степені вершини.

Зокрема, для вищенаведеної матриці, строка v4 дає суму елементів строки рівну нулеві, що відповідає deg(v4) =0 для графа G, відповідно до рис. 10.

5. Списки суміжності. Спосіб включає перерахунок всіх вершин, суміжних даній.

$\forall v \in V$ виконується $\Gamma(v) = \{v = V \mid (v_i, v_j) \in E\}$.

Наприклад, для графа G, показаного на рис. 10, маємо наступні списки суміжності

Γ(v1) = {v2, v3},

Γ(v2) = {v1, v3},

Γ(v3) = {v1, v2},

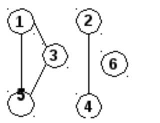
Γ(v4) = ∅,

Γ(v5) = {v6},

Γ(v6) = {v5}.

В конструюванні самим наглядним і зручним є матричний спосіб.

Приклад 3.1. Побудувати діаграму, матрицю суміжності та список неорієнтованого графа $G = (V, E)$, у якого $V = \{1,2,3,4,5,6\}$ та $E = \{(1,3), (1,5), (2,4), (3,5)\}$.



діаграма графа

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

матриця суміжності

1	3, 5
2	4
3	1, 5
4	2
5	1, 3
6	

список ■

Рис. 3.1. **Способи задання графа.**

В якості ілюстрації розгляданих топологічних понять, розглянемо резистивне коло , котре показане на малюнку 1.1.

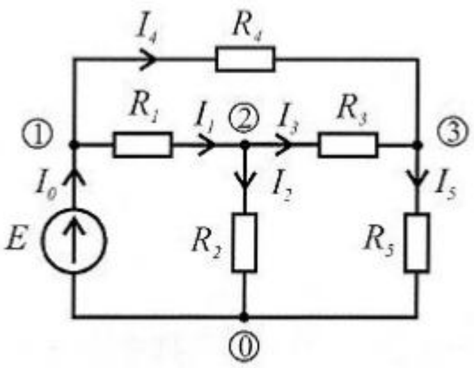


Рис. 1.1

Граф кола показаний на рис. 1.2. На рис. 1 показано дерево графа, яке утворене гілками 1,2 і 3. Гілки показані на рис. 1 пунктиром, - являються хордами.

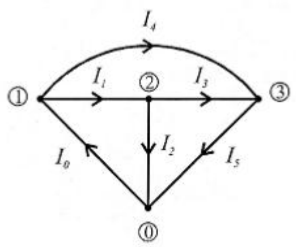


Рис. 1.2

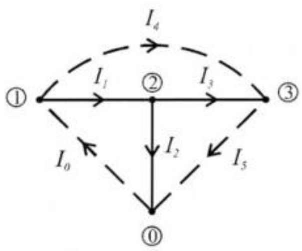


Рис. 1.3

Матриця інцеденцій [A] – це таблиця котра має в собі n_y строк , й n_B стовбців. Кожна строка відповідає вузлу, а кожен стовбець – гілці графа. Є гілка з номером j котра направлена від вузла I то в i-тій строці й j-ому стовбці записується +1. Якщо j-а гілка направлена до вузлу i , то в i-ій строці й у j-ому стовбці записується -1. Всі інші елементи матиці інцеденцій рівні нулю.
Матриця інцеденцій дє повний опис направленого графа. За допомогою матриці інцеденцій зручно записати рівняння по першому закону Кірхгофа у матричному вигляді :

$$[A][i^{в}]=0\,.$$

Припустимо , що ми знаємо напругу всіх вузлів ланцюга відносно базисного вузла. Тоді напруга гілок кола буде знайдена за допомогою матриці інцеденцій:

$$[u^{в}]=[A]^{t}[u^{у}].$$

Матриця інцеденцій для ланцюга , показаного на рис.1.1, має вигляд:

$$[A]=\begin{bmatrix} -1 & 1 & 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & 1 \end{bmatrix}.$$

Матриця головних контурів [B] описує з'єднання гілок, котрі входять в незалежні контури. Вона представляє таблицю розміром n_k x n_b . Кожна строка відповідає головному контуру, котрий з'явився приєднанням до дерева гілки зв'язку, а стовбець – гілці. Якщо i-а гілка входить в j-ий контур, то елемент $b_{ji}=\pm1$. Інші елементи матриці головних контурів рівні нулю.

Матриця головних контурів являє собою матрицю коефіцієнтів системи рівнянь, записаних в відповідності до другого закона Кірхгофа для головних контурів. Ось чому система рівнянь по другому закону Кірхгофа в матричній формі має вигляд :

$$[B][u^{в}]=0\,.$$

Матриця головних контурів, відповідає графу, показаному на рис. 1.3, й має вигляд :

$$[B]=\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 & 0 & 1 \end{bmatrix}.$$

Стовбець , котрий відповідає гілці зв'язку , має в собі тільки один не нулевий елемент, рівний +1. Кожна гілка дерева входить в два не нулевих елемента, рівних +1 й -1.

Топологічні матриці дозволяють формалізувати запис рівнянь по законам Кірхгофа, що дуже важливо з точки зору машинних розрахунків.

Активация Windows
Чтобы активировать Windows, пе
раздел "Параметры"

Топологічні матриці дозволяють формалізувати запис рівнянь по законам Кірхгофа, що дуже важливо з точки зору машинних розрахунків. Розглянемо алгоритм формування матриці інцеденцій.

Вихідні данні про коло представляються у вигляді таблиці з'єднань. Інформація про кожну гілку зберігається у виді триплета k , i , j . Тут k – номер гілки, i , j – номер вузлів підключення . Формування матриці інцеденцій виконується в такому порядку .

На першому кроці всі елементи матриці рівні нулю:

$$[A] = 0.$$

Потім переглядається таблиця з'єднань. Якщо k – а гілка, включена між вузлами з номерами гілки i й j , то $a_{ik}=+1$, $a_{jk}=-1$.

Формування матриці інцеденцій закінчується , коли буде досягнутий кінець списку гілок.

Алгоритм формування матриці інцеденцій являється найбільш простим. Процедура отримання матриці головних контурів являється складнішою, оскільки з початку необхідно обрати дерево графа.

Якщо граф має багато вершин і ребер, то він втрачає оглядовість. З таким графом важко працювати: у підрахунках трапляються помилки. Тоді відображають граф у вигляді матриці М. Найбільш поширеними є такі матриці графів: матриця суміжності відповідно вершин і ребер, матриця інциденті, матриця доступності, матриця відстаней. Накреслимо два графи (рис.3.18), відобразивши їх матричні еквіваленти

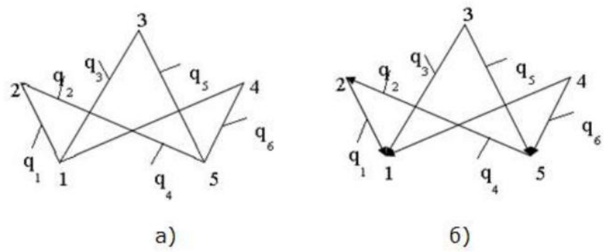


Рисунок 3.18 - Неорієнтований (а) та орієнтований (б) графи з різними матрицями суміжності

Активация Windo
Чтобы активировать W
раздел "Параметры".

Матриця суміжності вершин графа R - це квадратна матриця, в якій рядки і стовпчики відповідають вершинам графа. Порядок матриці збігається з кількістю вершин. При заповненні матриці на перетині рядка і стовпчика, що відповідають суміжним вершинам, ставлять 1; решту комірок заповнюють нулями. Мсм вершин для графа(рис. а) представлена такою таблицею:

$$\text{Мсм в} = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Матриця суміжності ребер графа Мсм.р - це квадратна матриця, в якій рядки і стовпчики відповідають ребрам графа. Порядок матриці збігається з кількістю ребер. При заповненні матриці на перетині рядка і стовпчика, що відповідають суміжним ребрам, ставлять 1; решту комірок заповнюють нулями. Мсм ребер для графа(рис. б) представлена такою таблицею

$$\text{Мсм.р}=\begin{bmatrix}0&1&1&1&0&0\\1&0&0&0&1&1\\1&0&0&1&1&0\\1&0&1&0&0&1\\0&1&1&1&0&1\\0&1&0&1&1&0\end{bmatrix}$$

Матриця інциденції дуг графа Мін. – це ортогональна матриця, в якій стовпці відповідають вершинам графа, а рядки – ребрам. Матрицю заповнюють так: на перетині стовпчика, що відповідає певній вершині, з рядком, що відповідає ребру, яке з цієї вершини виходить, ставлять -1; з рядком, що відповідає ребру, яке в цю вершину входить, ставлять +1; з рядком, що відповідає ребру, яке неінцидентне даній вершині – 0. Мін. дуг для графа (рис. б) представлена такою таблицею

$$\text{Мін}=\begin{bmatrix}1&-1&0&0&0\\0&1&0&0&-1\\1&0&-1&0&0\\1&0&0&-1&0\\0&0&-1&0&1\\0&0&0&-1&1\end{bmatrix}$$

Матриця інциденції ребер графа Мін. – це ортогональна матриця, в якій стовпці відповідають вершинам графа, а рядки – ребрам. Матрицю заповнюють так: на перетині стовпчика, що відповідає певній вершині, з рядком, що відповідає ребру, якому ця вершина інцидентна, ставлять + 1; решту комірок заповнюють нулями. Мін. ребер для графа (рис.а) представлена такою таблицею

$$\text{М'ін}=\begin{bmatrix}1&1&0&0&0\\0&1&0&0&1\\1&0&1&0&0\\1&0&0&1&0\\0&0&1&0&1\\0&0&0&1&1\end{bmatrix}$$

Матриці суміжності та інциденції визначають графи з точністю до ізоморфізму. За цими матрицями можна точно накреслити ізоморфні їм графи.

Матриця доступності графа Мд. – це квадратна матриця, в якій рядки і стовпчики відповідають вершинам графа. Матрицю заповнюють так: на перетині рядка і стовпчика ставлять 1, якщо j-та вершина доступна з і-ї, решту комірок заповнюють нулями. Для зв'язаного неорієнтованого графа всі елементи Мд рівні 1. У Мдорографа частина елементів може дорівнювати нулеві. Це засвідчує така матриця, ізоморфна графу (рис. б)

$$\text{Мд}=\begin{bmatrix}1&1&1&1&1\\0&1&1&1&1\\0&0&1&0&0\\0&0&0&1&0\\0&0&1&1&1\end{bmatrix}$$

Часто елементами Мд є топологічні відстані від і-тої до j-тої вершини, тобто кількість ребер за найкоротшим ланцюгом.

Матриця відстаней графа Мв - це квадратна матриця, в якій рядки і стовпчики відповідають вершинам графа. Її елемент показує відстань від і-тої до j-тої вершини, виміряну кількістю ребер (дуг) за найкоротшим маршрутом.

Якщо в орографі шлях від вершини відсутній, то відстань дорівнює нескінченності. Матриця відстаней орографа (рис. 3.18, б) має такий вигляд:

$$M_в=\begin{bmatrix} 0 & 1 & 1 & 1 & 2 \\ \infty & 0 & 2 & 2 & 1 \\ \infty & \infty & 0 & \infty & \infty \\ \infty & \infty & \infty & 0 & \infty \\ \infty & \infty & 1 & 1 & 0 \end{bmatrix}$$

Аналогічно визначається матриця відстаней неорієнтованого графа, хоч тут нескінченні відстані відсутні.

Матриця циклів графа Мц. У цій матриці кожному незалежному циклові відводиться рядок, а кожному ребру – стовпчик. Матрицю заповнюють так: на перетині стовпчика, що відповідає певному ребру, з рядком, що відповідає циклу, який має дане ребро, ставлять 1; решту комірок заповнюють нулями.

На відміну від матриць суміжності і матриць інциденції, матриця циклів не визначає графа з точністю до ізоморфізму.

Дерева в теорії графів

Найпростіші зв’язні графи на яких базується цілий клас в теорії алгоритмів задач носить назву – Дерева. Дерево G= <V, E> це зв’язний ациклічний (без циклів) граф. З того, що граф G – зв’язний, витікає, що між будь-якими його вершинами є маршрут типу v1e1v2e2....envn. Однак, оскільки граф G – ациклічний, то він завжди простий. Такий граф схожий на дерева всі гілки якого зв’язані, але не мають циклів, див. рис. 14.

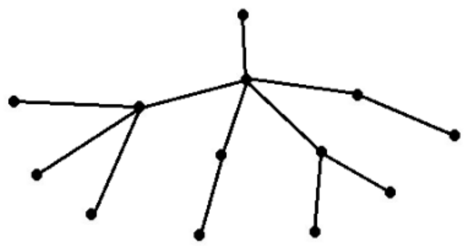


Рисунок 14 – Дерево G

Оскільки на основі зображення дерева не можливо встановити властивості, тому сформулюємо і доведемо теорему про характерні властивості дерева в теорії графів.

Теорема: наступні визначення справедливі і рівносильні:

1) G – дерево – тобто, зв’язний і ациклічний граф.

2) Між будь-якими двома вершинами графа G існує єдиний і простий ланцюг. Тобто, вершина і ребра не повторюються.

3) G – зв’язний і кількість ребер на одне менше кількості вершин.

4) G – ациклічний граф і кількість ребер на одне менше кількості вершин.

Граф G= <V, G> називається планарним, якщо можна так зобразити його на площині, що перетини його ребер на зображенні будуть лише у вершинах. Або по іншому: планарний граф ізоморфний деякому плоскому графу, а плоский граф – це граф у якого ребра на зображенні не перетинаються. Ізоморфний граф – це граф, отриманий з даного шляхом зміни його зображення на площині. На рис. 20 зображено планарний граф, що має чотири вершини і його ребра перетинаються лише в цих вершинах.

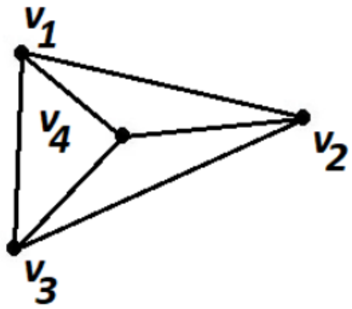


Рисунок 20 – Планарний граф

На наступному рис. 21 зліва показано граф на вершинах {a,b,c,d}, який є планарним, оскільки граф зліва ізоморфний плоскому графу на вершинах {v10,v20,v30,v40}, зображеному справа

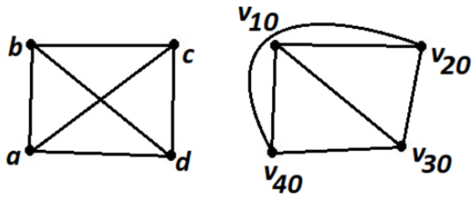
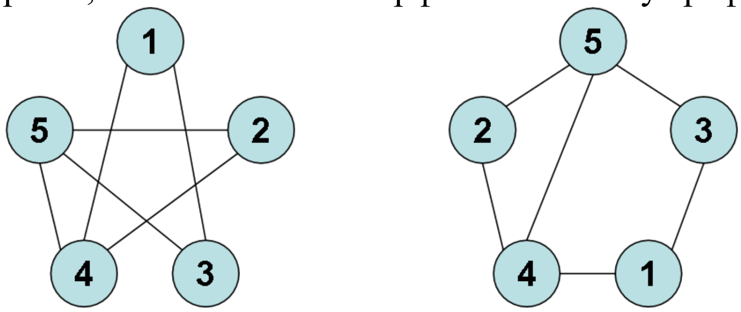


Рисунок 21. Планарний граф та ізоморфний йому плоский граф (справа)

Граф $G=(V,E)$ називається **планарним**, якщо його діаграму можна зобразити („укласти”) на площині чи сфері так, щоб ребра перетиналися лише у вершинах. **Плоским** називається граф, діаграма якого містить перетини ребер лише у вершинах. Таким чином планарний граф є ізоморфним деякому плоскому графу. На рис. 3.8 граф (а) не є плоским, але є планарним, оскільки він ізоморфний плоскому графу (б).



(а) (б)

Рис. 3.8. **Приклад планарного та плоского графа.**

Поняття графа дозволяє структурувати інформацію та алгоритмізувати цілий ряд типових задач. Саме тому існує цілий ряд алгоритмів які працюють з інформацією, що представлена у вигляді графів. Або їх ще називають алгоритми на графах. Зокрема, кожен раз коли ви відкриваєте файл на комп’ютері, останній запускає алгоритм пошуку в пам’яті адреси файлу. При цьому пошук виконується з використанням алгоритму на графах – пошуку бінарному дереві. Розглянемо цей та ряд інших алгоритмів детальніше.

Під **обходом** графа будемо розуміти деяке систематичне перелічення його вершин та/або ребер. Серед усіх обходів графа найбільш відомими та вживаними є пошук у *глибину* та *ширину*. Найбільш зручно виконувати ці обходи у випадку, коли граф задано списком.

Алгоритм пошуку на графі в ширину, або ще називають – обхід графа в ширину.

Задачею алгоритму є пошук певної інформації у структурованих даних, поданих у вигляді графа. Найпростішим прикладом роботи алгоритму пошуку у ширину є колекціонування (марок, значків і т.д.). Алгоритми дослідження графа шар за шаром, починаючи з вершини S, показано на рис. 63. Припустимо, що ми, знаходячись на нульовому рівні у дереві,шукаємо марку зі слоном, якої в нас немає. В першу чергу шукаємо цю марку серед близьких знайомих і друзів, які знаходяться найближче до нас(рівень 1), далі – серед знайомих цих близьких знайомих (рівень 2), далі серед інших знайомих і врешті-решт на деякому n рівні ми знаходимо шукану марку. Або припиняємо пошук,обійшовши всі n рівнів знайомих

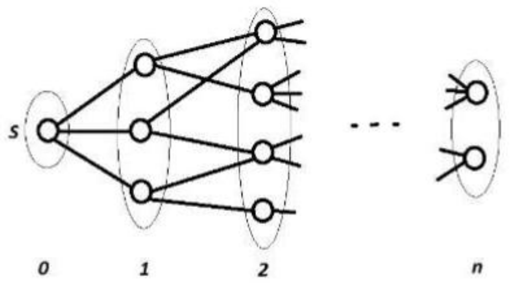


Рисунок 63 – Обхід графа в ширину

Алгоритм 3.2. **Побудова обходу графа у ширину.**

Спочатку всі вершини, окрім поточної вершини з номером 1 рівня $k=1$ є не поміченими.

Загальний крок. З кожної вершини рівня k почергово переходимо до усіх суміжних з нею непомічених вершин (рівня $k+1$) за зростанням їх номерів та помічаємо кожну з цих нових вершин. Далі знову виконуємо загальний крок.

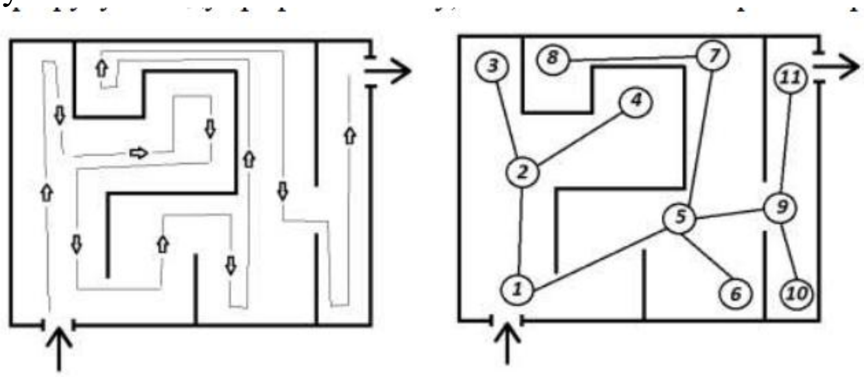
Алгоритм <u>закінчує</u> свою роботу у випадку, коли для кожної вершини рівня q не можна побудувати вершину рівня $q+1$.

За аналогічним алгоритмом відбувається збирання кубика Рубика, збирання пазлів та інші логічні ігри. Алгоритм пошуку в ширину (англ. breadth-firstresearch(BFS) використовують також для:

- Web серфінгу,
- пошуку друзів у соціальних мережах,
- доступу до мережі Internet,
- − обрахунку кон'юнкції,
- − перевірки моделей в теорії автоматів та ін.

Алгоритм пошуку в глибину.

Наступним розглянемо популярний алгоритм пошуку на графі в глибину, або ще називають – обхід графа в глибину(англ. Depth-First Search (DFS). Задачею алгоритму є пошук певної інформації у структурованих даних, поданих у вигляді графа. В цілому, алгоритм пошуку у ширину розглядає простий зв’язний граф G= <V, E>. Задачею алгоритму є побудова маршруту під початкової і до всіх вершин графа, обхід графа. Найпростішим прикладом роботи алгоритму є пошук виходу з лабіринту – якщо ми можемо рухатися вперед, ми рухаємося вперед, шукаючи вихід. В іншому разі повертаємося на перехрестя і рухаємося за іншим маршрутом. Однак, на відміну від попереднього алгоритму, пошук йде не за рівнями, а за маршрутом.



а) б)

Рисунок 65 – Лабіринт а) та граф шляхів у ньому б)

Алгоритм 3.1. **Побудова обходу графа у глибину.**

Спочатку всі вершини не помічені, поточною є вершина з номером 1.

Крок 1. Помічаємо поточну вершину і переходимо до кроку 2.

Крок 2. З поточної вершини, якщо це можливо, переходимо до непоміченої вершини з мінімальним номером. Цю вершину робимо поточною та переходимо до кроку 1. Якщо ж цей крок виконати неможливо (тобто всі вершини, суміжні з поточною є поміченими), то переходимо до кроку 3.

Крок 3. Повертаємося до тієї вершини, з якої ми перейшли до поточної. Цю вершину вважаємо поточною та переходимо до кроку 2.

Алгоритм закінчує свою роботу у випадку, коли поточною є вершина з номером 1 та крок 2 виконати неможливо.

Проілюструємо хід виконання обходів у глибину та ширину на наступному прикладі.

Приклад 3.3. Побудувати обходи у глибину та ширину для графа, заданого списком:

1	3, 5, 6
2	7
3	1, 6, 7
4	5, 7 .
5	1, 4, 7
6	1, 3
7	2, 3, 4, 5

Результат побудови обходів у глибину та ширину за алгоритмами 3.1 та 3.2 для цього графа є таким:

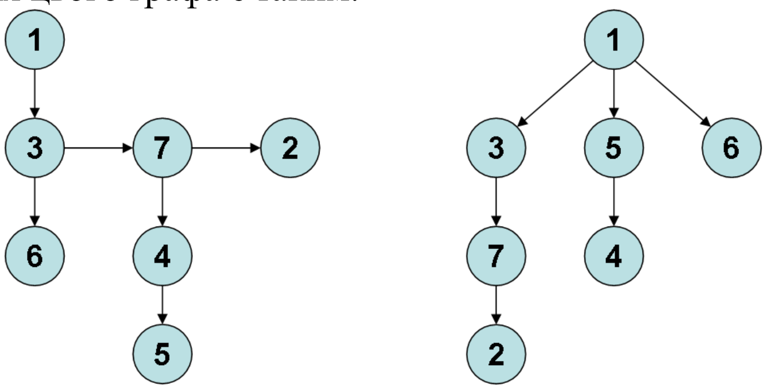


Рис. 3.9. **Обходи у глибину (ліворуч) та ширину для графа з приклада 3.3.**

Зауважимо, що при побудові обходу у глибину поточними були вершини
1, 3, 6, 3, 7, 4, 2, 4, 7, 5, 7, 3, 1; при побудові обходу у ширину поточними були вершини 1, 3, 5, 6, 7, 4, 2.

Приклад 3.2. Визначити, які графи з наведених на рис. 3.2. є ізоморфними між собою.

а) б) в)

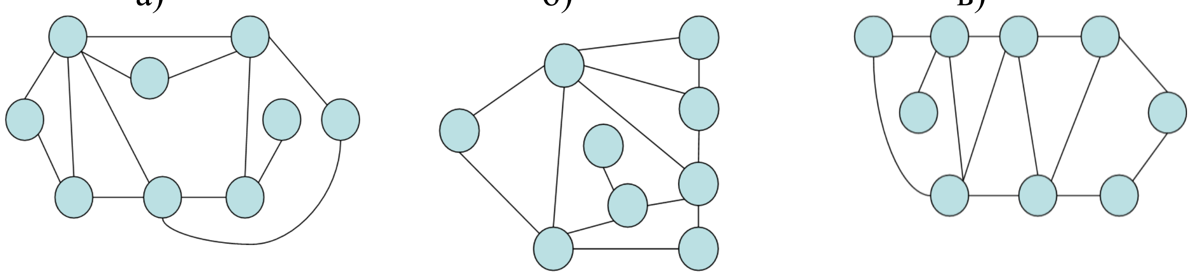


Рис. 3.2. **Графи (а), (б) та (в).**

Довільним чином нумеруємо вершини графа (а).

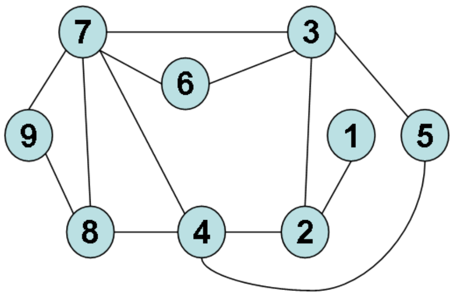


Рис. 3.3. **Нумерація вершин графа (а).**

У відповідності до цієї нумерації будемо нумерувати вершини графа (б).

Оскільки граф (б) має лише одну висячу вершину, то ми їй надамо номер 1. Ця вершина в графі (а) суміжна з вершиною 2, тому ми можемо однозначно надати номер 2 в графі (б):

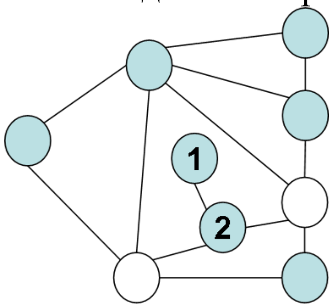


Рис. 3.4. **Однозначна нумерація вершин 1 та 2 графа (б).**

В графі (а) вершина 2 суміжна ще вершинам 3 та 4, які мають степінь 4. Такі самі вершини існують в графі (б) (вони виділені білим кольором). Треба з'ясувати яка вершини в графі (б) відповідають вершинам 3 та 4 графа (а). В графі (а) вершина 3 суміжна вершинам зі степенями 5, 3, 2, 2, а вершина 4 – 5, 3, 3,2. Тому ми можемо однозначно надати номери 3 та 4 виділеним вершинам графа (б).

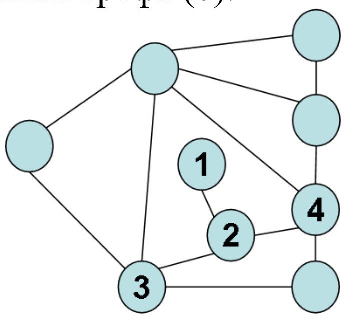


Рис. 3.5. **Однозначна нумерація вершин 1, 2, 3 та 4 графа (б).**

Тепер ми можемо однозначно надати номери 8 (суміжна з вершиною 4 та має степінь 3); 7 (суміжна з вершинами 3, 4 та має степінь 5); 5 (суміжна з вершинами 3, 4 та має степінь 2); 6 (суміжна з вершиною 3 та має степінь 2); 9 (суміжна з вершинами 7, 8 та має степінь 2):

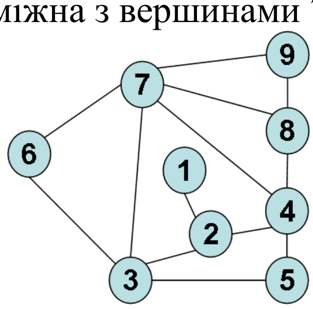


Рис. 3.6. **Остаточна нумерація вершин графа (б).**

Таким чином одержали, що граф (а) є ізоморфним графу (б).

Визначимо тепер, чи є ізоморфними графи (а) та (в). Вершини графу (а) нумеруємо так, як і в попередньому випадку. Так саме, ми можемо однозначно надати номери 1 та 2 вершинам графу (в):

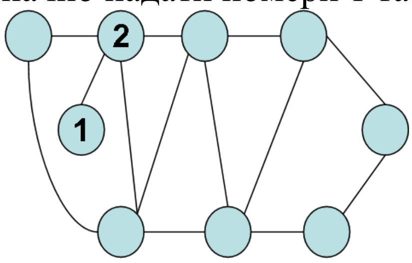


Рис. 3.7. **Однозначна нумерація вершин 1 та 2 графа (в).**

Проте в графі (а) вершина 2 має степінь 3, а у графі (в) – степінь 4. Тому графи (а) та (в) не є ізоморфними.

Приклад

1	3, 4, 6
2	3, 5, 6
3	1, 2
4	1, 5, 6
5	2, 4, 6
6	1, 2, 4, 5

Побудуємо дерево досяжності для цього графа із коренем у вершині 1 (рис 3.11).

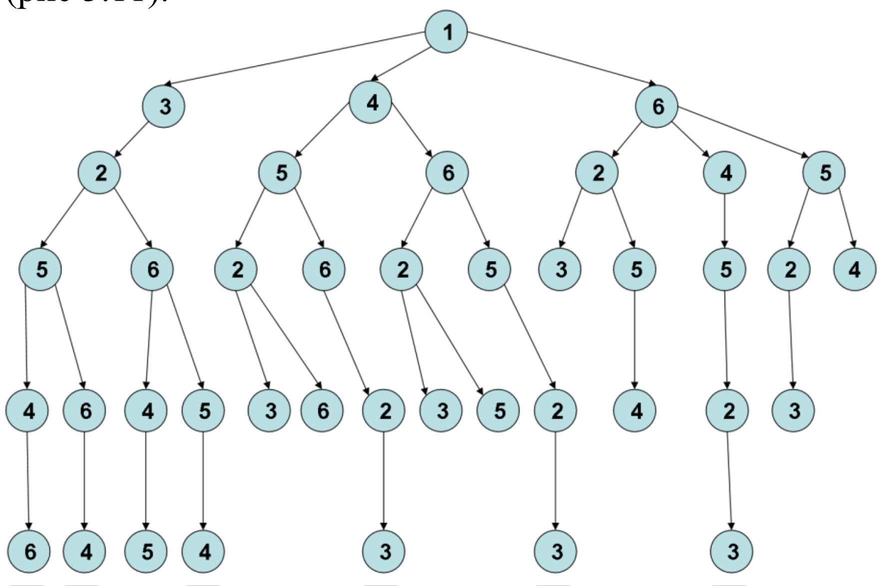


Рис. 3.11. **Дерево досяжності для графа з приклада 3.5.**