КДМ: Лекція 7. Колос К. Р.

Сьогодні розпочинаємо новий модуль, нову тему, – все нове.

ВСТУП ДО ТЕОРІЇ ГРАФІВ

Ви, мабуть, чули, що таке «граф»? Хтось десь там, колись… Це точки, це лінії між точками… все таке…

Насправді графи – це досить потужний засіб візуалізації процесів. Те, що можна спростити до того, що вам важливе і досліджувати саме те, що вам важливе шляхом розглядання малюнків. Адже людський мозок, як ми пам'ятаємо, він дуже пристосований до розглядання малюнків і розпізнавання того, що там робиться.

Застосовується теорія графів усюди: хімічні процеси, кристалографія, теорія прийняття рішень, комбінаторні оптимізації, діаграми Фейнмана у фізиці, транспортні задачі, логістичні задачі, компілятори тощо. Саме те, що це дуже просто… що там може бути складного? Ну точки, ну лінії між точками (рис. 1)…

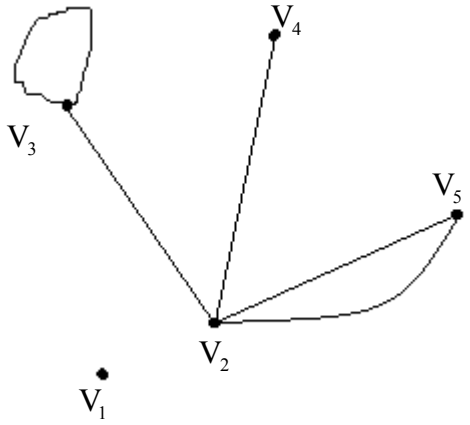


Рис. 1

– можна трактувати як міста і дороги,

– можна трактувати як обчислювальні елементи на електронній платі і зв'язки між ними,

– можна трактувати як ваш родовід: пам'ятаємо множину людей і відношення батьківства до синівства тощо…

Іноді виникає думка, що це дуже просто… щось на рівні школярів… наступні лекції я спробую зробити так, щоб ви переконались в іншому – що теорія графів – це дуже потужна математика.

Знову таки… коли ми розпочинаємо вивчати нову тему – ми розпочинаємо з чого?

1) з означень – 2) способів представлення – 3) операції – це все ми за цю лекцію встигнемо – а потім – 4) те, що з цим всім можна робити.

Неорієнтований граф

Неорієнтованим графом ми називаємо впорядковану пару з двох множин V та E: $G = \langle V, E \rangle$, де $V \neq \emptyset$ (*це не порожня множина*), яку ми назвемо «множиною вершин» (Чому V? Тому, що «вершина» англ. vertex). А множина $E \subseteq V^{(2)}$ – це множина невпорядкованих пар вершин, яку ми звемо «ребра» – множина ребер (ребра – англ. edge).

Символ $V^{(2)}$ (якщо у вас просто V^2 – це декартовий добуток – це множина впорядкованих пар); якщо ми ділимо її відношенням розпорядкування (тобто пари (b,a) та (a,b) будемо вважати одним і тим самим), то буде <u>відношення розпорядкування</u>. Тому ми кажемо, що цей граф неорієнтований, тому що якщо дві вершини поєднані ребром, то нам не важливо в який бік (це для нас однакова ситуація). *Потім у нас будуть орієнтовані графи – нам це буде важливо*.

(*Дописуємо назви вершин*) Це в нас якісь вершини: V_1, V_2, V_3, V_4, V_5, – і якісь ребра. Бачимо: тут у нас є ізольовані вершини, є парні ребра, є петлі – є всі ситуації. Всі ці означення будемо вводити, коли вони нам знадобляться.

Здається, тут все зрозуміло…

Насправді таких графів ще класифікувати багато. Ми будемо досліджувати найпростіший випадок.

Типи графів та їх узагальнення

1) Простий граф – це граф, який <u>не має парних ребер</u>. На рис. 1 – непростий граф, бо там є два ребра, що поєднують одні і ті ж самі вершини; в простому графі це забороняється.

Говорячи формальною мовою: в звичайному графі «$\subseteq$» може бути «мультимножиною», у простому гафі це просто множиною – воно забороняє повторне використання ребер.

2) простий граф без петель (*як це випливає з його назви?* ☺) – це простий граф, який додатково не має ось таких петель (*показати петлю на рис. 1*), тобто ребра, які виходять і входять в одну і ту ж саму вершину в ньому забороняються.

3) Псевдограф – це граф, який має петлі. І він, в залежності від контексту, може бути простим або непростим.

Далі… нічого не сказали про множину вершин, нічого не сказали про множину петель… Для нас це якийсь довільні множини, окрім того, що множина вершин повинна бути не порожня. І відповідно, якщо множина вершин є скінченною, то множина ребер є скінченною, то ми кажемо, що граф є скінченним, тобто його можемо обмежити навіть візуально – за допомогою візуалізації.

4) Скінченний граф: (*якщо потужність* $< \infty$, *то це є скінченне натуральне число)* $|V| < \infty, \quad |E| < \infty$.

В інших випадках ми кажемо, що цей **граф є нескінченним**.

Тобто це може бути граф, в якому нескінченна кількість вершин, наприклад, всі натуральні числа; може бути граф, в якому нескінченна кількість ребер – якщо дозволити нескінченну кількість ребер, то їх можна вибирати нескінченну кількість.

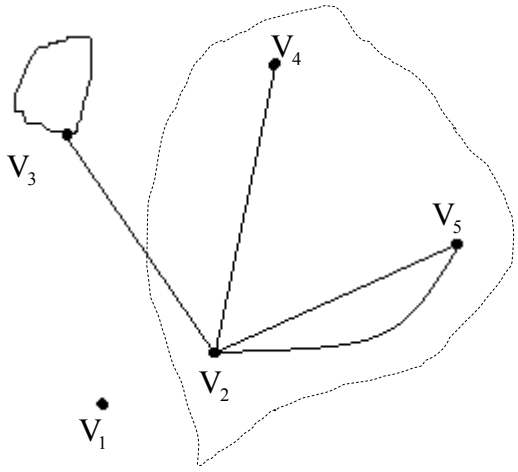


Рис. 2

5) Мультиграф – це граф, в якому дозволяються парні ребра, а також дозволяються так звані «сукупні вершини».

(*Що таке парні ребра – зрозуміло*).

Сукупна вершина – це множина вершин, яка розглядається як окрема вершина.

Скажімо, я можу взяти ці 3 вершини – і розглядати їх як одну (рис. 2). Але насправді то не 1 вершина, а множина вершин.

Відповідно там є певні засоби. Така ситуація виникає, коли, наприклад, розв'язуємо задачі класифікації: у вас є множина об'єктів, потім ви їх поєднуєте у певні класи і зв'язки між цими об'єктами вони скупочуються у зв'язки між класами. Тоді та ситуація… оператор мультиграфів природнім чином розглядається.

І тут, якщо казати формально, множина ребер є підмножиною і розглядаємо невпорядковані пари таких множин: $E \subseteq \left(2^V\right)^{(2)}$.

І нарешті…

6) Гіперграф – це конструкція, в якій ребро не є поєднанням пари вершин, а є поєднання трійки вершин, четвірки вершин, п'ятірки вершин тощо (ребра поєднують k вершин, де $k \geq 3$). Тобто це певне узагальнення, яке зветься «гіперплощиною». Там ми виходимо в n-вимірний простір, там ребро – це багатомірна площина, яка поєднує ці вершини. І зрозуміло: $E \subseteq V^{(k)}$, – множина ребер представляється як підмножина впорядкованих k-тов.

Приклад гіперграфа. Берете кубик Рубика, його грані – це ребра гіперграфа. Адже кожна сторона поєднує 4 вершини.

Далі ми будемо розглядати лише скінченні прості графи без петель (якщо не буде наголошено іншого, але я тоді буду наголошувати).

Уявили собі «безмежний всесвіт» теорії графів? ☺

Способи представлення графів

Два способи у нас уже тут фігурують, але я їх витру ☺

1) Явний спосіб. Граф – це пара множин: V, E. Кожну множину можна подати у явному вигляді. Особливо, якщо це скінченний граф. *Так?*

Наприклад, у вас буде граф $\mathrm{G}=\langle \mathrm{V},\mathrm{G}\rangle$, що складається з пари множин V та E: $\mathrm{V}=\{\mathrm{a},\mathrm{b},\mathrm{c},\mathrm{d}\}$, $E=\{(a,b),(a,c),(b,d),(b,c)\}$ – тут просто перелічуються пари вершин, які поєднуються ребрами.

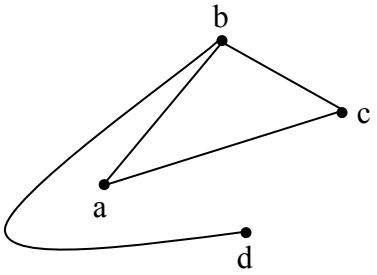


Рис. 3

2) Графічний спосіб. Тобто, я беру і малюю діаграму для графа $\mathrm{G}=\langle \mathrm{V},\mathrm{G}\rangle$: позначаю 4 вершини, заіменовую їх, і потім малюю ті ребра, які вказані у E: $E=\{(a,b),(a,c),(b,d),(b,c)\}$ (рис. 3).

Як малювати ребра? Як завгодно. Аби вони поєднували задані вершини.

Ребро (b,d) я намалювала ось так (як зображено на рис. 3), тому що якби я сполучила вершини b і d прямою, то таке ребро перетнуло б ребро (a,c) – і може виникнути хибна думка, що тут є вершина. Тому краще такого не робити. *Так?*

Зрозуміло, якщо я пронумерую першини в інший спосіб, то одержу іншу картину для того самого графа. У картинках вас ніхто не обмежує. Головне, щоб ці ребра тут були намальовані. (Графи – це засіб візуалізації. А для кого ви візуалізуєте? Для себе. Тому, як вам зручно, так і малюєте: хочете на площині, хочете у тривимірному просторі та ін.)

3) Матриця суміжності. *Я так розумію, що таке матриці ви вже знаєте. Ви вже їх перемножати вмієте. Це нам досить знадобиться наступної лекції.* Явний спосіб можна використовувати, якщо у вас мало вершин і мало ребер, то їх перелічення не буде викликати ніяких незручностей. Графічний – дуже зручний спосіб: все зрозуміло, але лише для людини. А в нас задача автоматичної обробки. Нам потрібно навчити комп'ютер. Комп'ютери не вміють із картинками працювати. Поки-що. Ще теорія розпізнавання образів тільки розробляється.

Так от. Одним із способів опису графів для комп'ютерів – це задання графа за допомогою матриці суміжності $\mathrm{A_G}$. Значить, це $\mathrm{m}\times\mathrm{n}$ матриця, де $|V|=n$ – це кількість вершин, $\mathrm{A_G}=\|a_{ij}\|$, де $i,j=\overline{1,n}$ (i,j пробігають значення від 1 до n). *Ви так матриці позначали? Якщо ні, то позначайте ось так*.

$\mathrm{a_{ij}}=\left[(V_i,V_j)\in E\right]=$ (*що таке дужки Айверсона всі пам'ятають? Якщо ось це:* (V_i,V_j) – *виконується, то маємо одиницю; якщо ж це не виконується, то маємо 0*).

Тобто у нас у матриці стоїть 1, якщо відповідні вершини поєднані ребром і стоїть 0, якщо відповідні вершини не поєднані ребром: $=\begin{cases}1, (V_i,V_j)\in \mathrm{E}\\ 0, (V_i,V_j)\notin \mathrm{E}\end{cases}$.

Скажімо, для отого графа (рис. 3), який ми там намалювали матриця суміжності буде якого розміру? 4×4, бо у вас 4 вершини. Можете їх над матрицею намалювати в тому порядку, в якому вам зручно. А далі бачимо: які у нас є ребра? Ребро (a,b) – малюєте 1 і тут (b,a) 1, бо (a,b) і (b,a) – це одне і те ж саме ребро. І т.д. У всіх інших місцях – 0.

$$\mathrm{A_G}=\begin{array}{c} \\ \mathrm{a}\\ \mathrm{b}\\ \mathrm{c}\\ \mathrm{d}\end{array}\begin{array}{c}\begin{array}{cccc} a & b & c & d\end{array}\\ \begin{bmatrix}0&1&1&0\\1&0&1&1\\1&1&0&0\\0&1&0&0\end{bmatrix}\end{array}.$$

Зауваження:

1) $\mathrm{A_G}$ **– симетрична.** У неорієнтованого графа матриця суміжності завжди симетрична (тобто вона відносно головної діагоналі симетрична – її можна транспонувати; *термін*

«транспонування»? Симетрія відносно діагоналі. Так? Тобто я її можу повернути – і одержу точно таку ж саму матрицю; це через те, що у нас ребра є невпорядковані; у вас (a,b) і (b,a) – це одне і те ж саме ребро, тому воно або в обох місцях є, або його немає);

2) матриця суміжності неорієнтованого графа A_G **визначає (зображає) відношення сусідства на вершинах**. Тобто це матриця бінарного відношення, заданого на множині V, яке поєднує вершини, які ми з'єднали ребром. Тому для нас графи є засобом візуалізації, зокрема, відношень. І відповідно, весь апарат теорії відношень можна застосовувати для аналізу графів. Це відношення буде симетричним через те, як ми ввели неорієнтовані графи.

Інший спосіб представлення матричного графа – це так звана «матриця інцидентності».

4) Матриця інцидентності B_G**.**

Якщо у матриці суміжності ставлять 1 там, де вершини поєднувалися ребром, то матриця інцидентності зв'язує вершини і ребра.

Тобто це буде в нас $n \times |E|$ – матриця прямокутна; рядки в неї в неї будуть нумеруватися вершинами, а стовпчики – ребрами: $B_G = \|b_{ij}\|, i = \overline{1,n},\ j = \overline{1,|E|}$ (j пробігає значення від 1 – до кількості ребер). Відповідно ребра в множині ребер потрібно певним чином пронумерувати, щоб коректно побудувати матрицю інцидентності. І ми кажемо, що

$$b_{ij} = \begin{cases} 1, \text{якщо вершина } v_i \text{ - кінець ребра } e_j \\ 0, \text{яакщ ні} \end{cases}$$

(це буде 1, якщо вершина v_i є кінцем ребра e_j; і 0, якщо ні).

У кожного ребра 2 кінця: з одного і з іншого боку. Нехай: $E = \{(a,b),(a,c),(b,d),(b,c)\}$

це ребро e_1 e_2 e_3 e_4

І я беру матрицю

$$B_G = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} e_1 & e_2 & e_3 & e_4 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

Далі дивлюся: ребро e_1 поєднує *a та b*. Значить в нього кінці будуть *a та b*. А *c та* d до нього ніякого відношення не мають. І т.д.

Маємо таку матрицю інцидентності. Взагалі кажучи, термін «інцидентність»: тобто ми кажемо,, що вершини є інцидент ними ребру, якщо вони є його кінцем. *Ми ж пам'ятаємо: якщо математики не видумають чогось страшного для означення чогось простого і зрозумілого, то день даремно пройшов…* ☺

Які у нас властивості матриці інцидентності очевидні?

1. Скільки буде одиничок в кожному стовпчику? Дві. У вас ребро поєднує лише 2 вершини, тому в кожному стовпчику буде лише дві 1, не залежно від розміру матриці. (У графі з петлями є певні застороги… скажімо, в деяких випадках треба малювати не одиничку, а двійку, бо це ребро і виходить і входить в цю вершину одночасно…це для того, щоб зійшлися обчислення, які будуть з цими матрицями працювати; а в деяких випадках взагалі забороняються петлі в матрицях інцидентності малювати. Тому ми розглядаємо самий простий випадок – коли цього всього не виникає).

Отже, **кожен стовпчик матриці** B_G **містить рівно дві одиниці**.

2. А в рядку? Дивіться: скільки ребер виходить з вершини *a* (див. рис. 3)? Два. І в матриці інцидентності в цьому рядку повинно стояти дві одинички (для кожного ребра, яке виходить з вершини *a*). Тобто **кількість одиничок в рядку матриці інцидентності** B_G

дорівнює кількості ребер, які виходять з заданої вершини. Це число зветься степенем вершини. Формально ми його введемо на наступній лекції.

І останній спосіб

5) Списки суміжності.

Я для кожної вершини $\forall u \in V$, яка пробігає множину вершин, просто перелічую усі вершини, з якими вона поєднана ребрами. Тому я ввожу спеціальну множину (гама від у; Γ – це велика грецька літера «гама»):

$\Gamma(u) = \{v \in V \mid (u,v) \in E\}$ – це множина всіх таких вершин v, що ребро (u,v) належить множині ребер.

Знову таки: для цього графу (див. рис. 3) скільки буде списків суміжності? 4 (за кількістю вершин).

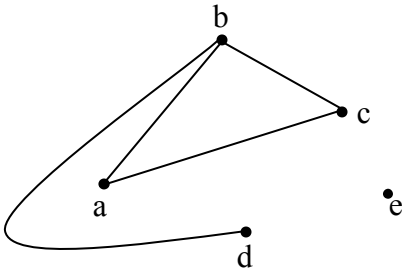
(список суміжності вершини *a? a поєднується з b, поєднується з c; тому це буде множина b та c*).

$\Gamma(a) = \{b, c\}$

$\Gamma(b) = \{a, c, d\}$

$\Gamma(c) = \{a, b\}$

$\Gamma(d) = \{b\}$.



Якби в мене тут була ще одна ізольована вершина e, – в неї список суміжності є порожньою множиною. (*витерти нарисовану нову вершину*)

Для різних задач, для різних типів графів кожен з цих способів може бути ефективним чи неефективним. Тобто в різних випадках треба використовувати різні представлення.

Скажімо, якщо у вас граф є, так званий, «розріджений» (тобто в ньому дуже мало ребер), то матриця суміжності, як була $n \times n$, вона так і залишається $n \times n$, але містить дуже багато нулів; а списки суміжності стають такими: малесенькими, малесенькими. Тому, що, якщо ребер мало, то їх поєднань між вершинами також мало. Тому:

для розріджених графів (sparse graphs) зазвичай списки суміжності є більш ефективним представленням (але не завжди, але зазвичай).