Алгоритмічні операції над графами

Те, що ми розглядали перед цим – це були алгебраїчні операції, тобто над графами як парами множин.

Є ще, так звані **«алгоритмічні операції» – це те, що ми можемо витворяти з графами в застосуванні певних процедур**.

Зараз почну – і ви зрозумієте, що це таке.

По-перше, це може бути видалення вершин – ви берете і в графі знищуєте вершину і ребра, які з нею пов'язані.

1) видалення вершин (разом із інцидент ними ребрами), – тобто вершина не просто видаляється з графа, – вона видаляється з усіма ребрами, які з неї виходять.

Так само може бути операція «видалення ребра».

2) видалення ребра (взяли гумку і витерли це ребро з рисунку; вершини залишаємо), тобто видалення ребра **просто прибирає зв'язок, але не чіпає вершини**.

Відповідно є і зворотні операції – «внесення ребра».

3) внесення ребра (у вас були дві вершини – і ви між ними намалювали ребро) – так **додали до множини E ще одну пару**.

Може бути «внесення вершини».

4) внесення вершини (взяли і намалювали збоку ще одну точку – оце нова вершина вашого графа).

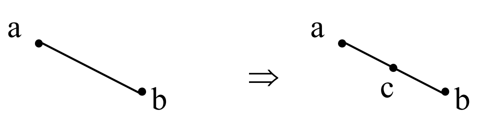
Якщо вам потрібно, щоб вона була з ребрами, то внесли вершину, а потім внесли потрібні ребра (бо ребра поєднують лише існуючі вершини).

Далі піде всяка екзотика…

Можна внести вершину у ребро

5) внесення вершини у ребро

Нехай у вас є ребро, що поєднує вершини a та b. А я беру і в його середину розташовую



нову вершину c. І було одне ребро, а стало два ребра, поєднаних через вершину c.

То ось така операція називається «внесення вершини до ребра».

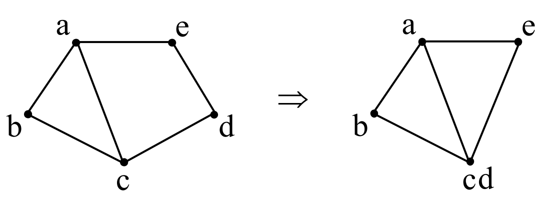
Далі… операція, яка називається «стягування ребра»

6) Стягування ребра

Дивіться за моїми руками: ось ребро – перша вершина і друга вершина. Беремо – і стягуємо.

Скажімо є в мене граф abcde. Я беру і роблю ось так (*проводжу ребро з вершини a до вершини c*). В що перетворюється граф?

Тобто я видаляю ребро *cd* – роблю *c* і *d* однаковою вершиною – і, відповідно, об'єдную



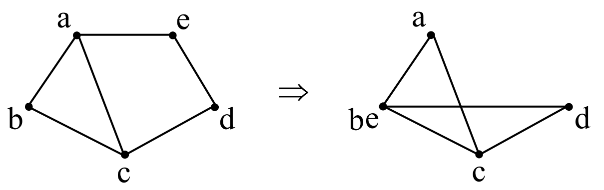
там ребра, які там є.

І одержую граф ось такий (див. рис. 6.2): тут буде вершина, яку я умовно назву *cd* і ребра будуть ось такі: *ab*, *acd*, *ae*, *bcd*, *ac*, *ecd*.

Тобто вершина a не поєднувалася ребром з вершиною *d*, але вона поєднувалася ребром із вершиною *c*, тому ребро від *a* до *cd* залишається.

7) Ототожнення вершин – це стягування ребра, якого може не існувати.

Тобто ми беремо дві вершини нашого графу і кажемо, що вони співпадають. Тобто, якщо було стягування ребра – ми просто ребро зліплювали в одне ціле, а ототожнення двох вершин – взяли дві вершини – і сказали, що вона одна.



Наприклад. Візьмемо цей граф (рис. 7.1). Візьмемо вершини *b* та *e*. І ототожнимо їх. Я повинна отримати 4 вершини: a, умовну вершину be, вершину c, вершину *d.*
Значить, всі ребра, які не стосувалися *b* та *e* залишаються. Тепер: у вас *b* була поєднана ребром із *a*, *b* була поєднана ребром із *c*. Це також залишається. *e* була поєднана ребрами із *a* та із *d*. І граф перетворюється на ось таку картинку (рис. 7.2). Ми практично взяли вершину *e* і перетащили її сюди (на вершину *b*) з усіма ребрами. (*– за резинучку потянули – і ось…одержали ось такий граф*)

Ось такі операції використовуються під час певних алгоритмічних застосувань графів, під час доведень певних теоретичних фактів. Коли нам чогось не вистачає, або щось зайве. Зайве можна прибрати. А того, чого не вистачає – ми будемо вносити.

Остання операція…

Для кожної вершини графу G ми вводимо поняття степеня.

8) Степінь вершини $\mathrm{v} \in \mathrm{V}$ – це просто кількість ребер, які виходять з цієї вершини або потужність відповідного списку суміжності: $\deg(\mathrm{v}) = |\Gamma(v)| = \#\{u \in V \mid (u, v) \in E\}$.

Для степенів має місце дуже простий факт, але дуже значущий для теорії графів.

Я не казала, але вся теорія графів виникла як математична забавка Леонарда Ойлера, який сидів і думав: «а чи можна пройти по Кінесберських мостах так, щоб пройти кожен міст один раз?»… *а потім понеслось ☺...*

Лема (про рукостискання).

Десь на минулій лекції я казала, що теорію графів можна сформулювати так, що її зрозуміють першокласники. Лема про рукостискання – це один з таких прикладів. Якщо вас тут сидить… скільки? 97 здається… трохи менше… ☺… ну і ви, коли прийшли, – ви один одному потиснули руки… так загальна кількість потиснутих рук – хто б з ким там не потискав руки – вона завжди буде парною.

А, якщо формально, лема про рукостискання каже нам, що якщо в довільному графі G просумувати всі степені вершин, то ми одержимо подвійну кількість ребер:

$$\forall \mathrm{G} \sum_{\mathrm{v} \in \mathrm{V}} \deg(\mathrm{v}) = 2 \cdot |E|.$$

Цей результат у нас очевидний. *Хто розуміє, що він очевидний?*

Оскільки степінь – це кількість ребер, які виходять із заданої вершини. Якщо ви підрахуєте всі степені кожної вершини – і додасте, то ви одержите загальну кількість ребер. А при цьому кожне ребро було підраховане двічі: з одного кінця – і з іншого кінця. Тому це буде загальна кількість ребер, в якому кожне ребро підрахували двічі. Тобто $2 \cdot$ *кількість ребер*. *Кінець доведення. Лему про рукостискання довели. Це дуже просто. Але…*

Наслідок 1. (ви ж пам'ятаєте… скільки тих наслідків з неї… ☺)

Скільки у вас може існувати вершин непарного степеня? Чи може у вас бути граф з 11 вершинами, у яких всі степені непарні? … Якщо у вас ось тут: $\sum_{\mathrm{v} \in \mathrm{V}} \deg(\mathrm{v})$, – 11 доданків і кожне з них непарне число – і ви це все додасте, то ви одержите непарне число. А за лемою про рукостискання у вас повинна сума бути парна: $\sum_{\mathrm{v} \in \mathrm{V}} \deg(\mathrm{v}) = 2 \cdot |E|$.

Тому **наслідок 1. В довільному графі кількість вершин непарного степеня є парною**. Інакше ця сума: $\sum_{\mathrm{v} \in \mathrm{V}} \deg(\mathrm{v}) = 2 \cdot |E|$, – не зійдеться.

Наслідок 2. Ми кажемо, що граф є k **-регулярним, якщо всі його степені дорівнюють** k**, тобто із кожної вершини виходить однакова кількість ребер:** $\forall \mathrm{v} \in V \ \deg(v) = k$.

Скажімо, граф K_n є $(n-1)$-регулярним, тому що з кожної вершини виходить $(n-1)$-е ребро. Простий цикл є 2-регулярним, тому що з кожної вершини виходить по 2 ребра. А колесо взагалі не є регулярним графом, тому що з одних вершин виходить по 3 ребра, а з центральної – багато.

Так ось, якщо у нас є k-регулярний граф на n-вершинах, то або k, або n є парним числом. Чому це так? Припустимо, що обидва ці числа є k і n є непарними. Тоді чому дорівнює ця сума: $\sum_{\mathrm{v}\in \mathrm{V}} \deg(\mathrm{v})$? Стосовно всіх степенів – всі степені дорівнюють k, доданків тут n, то ця сума буде: $\mathrm{n}\cdot\mathrm{k}$. Якщо обидва непарні, то і добуток їх непарний. А повинно бути парне! Протиріччя. …

У мене ще є 15 наслідків? Давайте ви їх будете розглядати на практичних. Добре. Дякую.

МАРШРУТИ У ГРАФАХ

Продовжуємо дослідження наших улюблених графів. Ми вже з вами ввели означення графа, способи їх представлення, алгебраїчні та алгоритмічні операції.

Ну а тепер те, що там є конструктивного, те, що нам потрібне.

Ми розглядаємо простий неорієнтований скінченний граф без петель: $\mathrm{G} = \langle V, E\rangle$, – який має n-вершин та певну кількість ребер, де кожне ребро є невпорядкованою парою вершин: $\mathrm{V} = \{\mathrm{v}_1, v_2, ..., v_n\}$, $E \leq V^{(2)}$.

Маршрутом (або шляхом) у графі (англ. walk, path, chain). Я попереджаю, що термінологія в теорії графів – вона трошки відрізняється: вітчизняна і закордонна. Отже, маршрутом у графі ми називаємо скінченну послідовність вершин, де у вас кожні дві вершини на маршруті поєднані ребром у графі (рис. 1), тобто є суміжними: $\forall i\ \mathrm{v}_\mathrm{i} \in V, (v_i, v_{i+1}) \in E$.

$$\mathrm{v}_1 \to v_2 \to \ldots \to \mathrm{v}_\mathrm{m}$$

Рис. 1

Це таке «страшне» визначення лише для того, щоб сказати, що якщо у вас є граф, а в ньому є якісь ребра, то маршрут – це просто шлях проходження по вашому графу в якомусь певному сенсі (рис. 2).

Тобто, a→b→c→d→b→d→b→a (a переходить в b, b переходить в c і т.д.) – це маршрут. *Ми гуляємо ребрами нашого графа*.

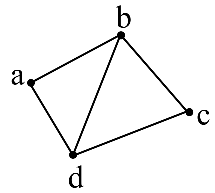


Рис. 2

А якщо, скажімо, я хочу зробити a→c (*a* переходить в *c*), то це не буде маршрутом, тому що в *a* та *c* не має ребра: a→c – не маршрут.

Відповідно довжина маршруту – це кількість ребер, які в ньому задіяні:

довжина маршруту = кількість ребер.

Якщо у вас якесь ребро повторюється, то воно рахується двічі.

Якщо у нашому прикладі ребро d→b і ребро d→b, то це два різних ребра на маршруті.

Далі… ми кажемо, що **маршрут є замкненим**, якщо в нього перша і остання вершина співпадають: $v_1 = v_m$.

Ланцюг (англ. *trail*) – це такий спеціальний маршрут, в якому немає повторів ребер.

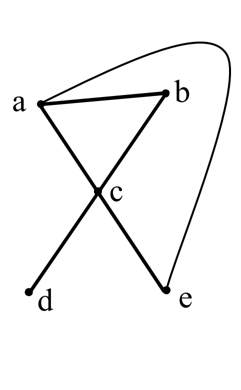
Простий ланцюг – це ланцюг, в якому і вершини не повторюються, окрім, можливо, першої та останньої. Тобто ланцюг може бути замкненим. І це дозволяється.

Цикл – це замкнений ланцюг.

Простий цикл – це простий замкнений ланцюг.

Візьмемо якийсь приклад. Скажімо, такий граф на п'яти вершинах. І, якщо я побудую шлях d→c→b→a→c→e, то це буде ланцюг. Ланцюг довжини 5. Але є повторювана вершина *c*.

Якщо я побудую щось на кшталт a→c→e, то це буде простий ланцюг, тому що і ребра всі різні, і вершини всі різні.



Якщо я потім ще добудую ребро з e→a: a→c→e →a, – одержимо простий цикл.
Чи є тут непростий цикл? … Здається немає.

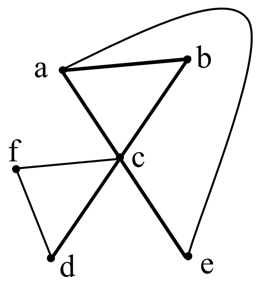


Рис. 4

Але, якщо я тут добудую вершину f і поєднаю її ребрами ось так (рис.4), то послідовність: d→f→c→b→a→c→d – це буде цикл, тому що ми починаємо з d (6), всі ребра різні, закінчуємо також у вершині d. Але у мене є повторювана вершина.

Тут поняття, які інтуїтивно зрозумілі, але їх є деяка кількість і я ними буду користуватися дуже часто, тому ви повинні розуміти, що я кажу «ланцюг», а що я розумію, коли кажу «простий ланцюг».

Твердження.

Є деякі прості властивості, які (знову таки) інтуїтивно зрозумілі, але їх, формально кажучи, треба доводити.

От ми сформулюємо їх три.

<u>Властивість 1.</u>

Якщо у вас є маршрут від вершини v_i – до вершини v_j, то у вас існує вершин від v_j до v_i. *Очевидно?...* Що це за маршрут? Треба всі вершини переписати у зворотному порядку. Оскільки у нас граф неорієнтований, то якщо є ребро від a→ до b, то є ребро b→a. *Для орієнтованих графів це буде невірне. Але про це поговоримо, коли у нас будуть орієнтовані графи.*

<u>Властивість 2.</u>

Якщо у вас існує маршрут з вершини v_i – до вершини v_j, то у вас існує простий ланцюг, який з'єднує ці вершини.

Тобто, якщо ми можемо хоча б якимось чином добратися до від однієї вершини – до іншої, то ми можемо це зробити не повторюючи ребра і навіть не повторюючи вершини.

Це ми зараз доведемо.

<u>Властивість 3.</u>

Довільний цикл розбивається у сукупність простих циклів. Скажімо, якщо ми розглянемо отой цикл (рис. 4), який ми намалювали: d→f→c→b→a→c→d, – то він розбивається у сукупність оцього цикла (d→f→c) і ось цього простого циклу (b→a→c). *Це ви будете самостійно доводити.*

Доведемо твердження 2.

➢ Нехай у вас існує послідовність $v_i \to v_2 \to v_3 \to \ldots \to v_k \to v_t \to v_j$, тобто у вас існує маршрут від v_i до v_j. І він пролягає через вершини v_2, v_3, v_4 і т.д.

Нехай цей маршрут не є простим ланцюгом. Тоді в ньому є, щонайменше, вершини, які повторюються. І нехай це будуть вершини v_k, v_t. Тоді, якщо вершина v_k і вершина v_t – одна і та ж вершина, то навіщо мені робити ось цей перехід: $v_k \to v_t$? То я будую одразу ось такий перехід: $v_i \to v_2 \to v_3 \to \ldots \to v_k \to v_t \to v_j$. Викидаю ось цю частину: $\to v_t$. А маршрут залишається маршрутом, тому що якщо тут є ребро $v_t \to v_j$, то й таке ребро: $v_k \to v_j$ – існує. Тому, що це одна і та ж сама вершина.

Бачимо, що новий маршрут є коротшим, тобто ребер у ньому менше. І по-друге, ось ця пара повторюваних вершин: v_k, v_t – зникає, залишається лише одна з них. *Зафіксуйте це, будь ласка.*

Тобто ми замінюємо частину від $v_k \to \ldots \to v_{k+1}$ на одне ребро $v_k \to v_{k+1}$. І в новому маршруті кількість ребер менша і на 1 (одиницю) менше повторюваних вершин.
Якщо це знову непростий ланцюг, то в ньому знову існує пара повторюваних вершин. Я знову можу викинути частину маршруту, побудувавши нове ребро зв'язку. І знову повторюваних вершин стане менше. Так я можу повторювати доти, доки в мене не залишиться ситуація, коли всі вершини різні. А тоді це буде простий ланцюг.
Властивість 3 доведіть самостійно… вона доводиться так само. Але, знов таки, ви повинні викласти все це формально коректно.
Всі вже знають, що таке матриці… так? Всі вже можуть множити матриці… так? Це добре. Це дуже добре.

Теорема (про степінь матриці суміжності).

Нехай у вас є скінченний простий без петель граф G і нехай A – це матриця суміжності цього графа: $G = \langle V, E \rangle, A = A_G$ – матриця суміжності (adjacency matrix). Матриця суміжності позначається – A_G, але я буду казати А, тому що в нас один граф G.
Нехай матриця A^k – це k-та степінь матриці суміжності, яка складається з елементів a_{ij}^k:
$A^k = \left\| a_{ij}^k \right\|,\ ij = \overline{1,n}$. Тоді ось це число a_{ij}^k – це кількість всіх <u>маршрутів</u> між вершиною v_i і вершиною v_j: $a_{ij}^{(2)} =$ кількість всіх маршрутів від v_i до v_j.
Я підкреслюю, що це власне <u>маршрути</u>, тобто там можуть бути довільні повтори ребер і вершин, але кількість буде коректна.
Доведення.
Як ми будемо доводити таке твердження? … за методом математичної індукції.

- Якщо $k = 1$, то матриця A^1 – це таки матриця суміжності: $A^1 = \left\| a_{ij} \right\|$. І за визначенням матриці суміжності – що є її елементами?

Тобто $a_i = \begin{cases} 1, \textit{якщо} \ \exists\ \textit{ребро}\ v_i \to v_j \\ 0, \textit{якщо не існує} \end{cases}$

Але ж ребро – це є маршрут, довжиною 1. Згодні? Тому, де $k = 1$ – ця матриця $A^1 = \left\| a_{ij} \right\|$ в точності описує кількість всіх маршрутів, довжини 1 між довільними вершинами.
<u>Нехай у нас твердження теореми справедливе для певного значення $k = m$. І розглянемо тоді маршрут, довжини $m+1$ між двома заданими вершинами: v_i та v_j.</u>
Схематично я це зображу ось так:
$\underbrace{v_i \to \ldots \to v_t}_{m} \to v_j$, v_t – вершина з передостаннім номером (не важливо яким).
Якою може бути вершина v_t? Да будь-якою. Тому загальна кількість маршрутів між ось цими вершинами: v_i та v_j – це кількість маршрутів від v_i – до v_t помножити на кількість маршрутів від v_t – до v_j. А маршрут[1] від v_t – до v_j – це або там є ребро, або там його немає. Так?
Тому я повинна розглянути всі можливі вершини v_t, які можуть бути передостанніми. А передостанніми можуть бути довільні вершини. Я повинна взяти кількість маршрутів від v_i – до вершини v_t і помножити їх на кількість маршрутів від v_t – до v_j, довжиною 1:

$$a_{ij}^{(m+1)} = \sum_{t=1}^{m} a_{it}^m \cdot a_{tj} \ .$$

Але що це за формула? Це формула для множення матриць A^m та матриці A.

[1] Маршрут – це кількість ребер

Бо в першому множнику ми пробігаємося по i-му рядку, а в другому множнику – по j-му стовпчику. Перемножуємо, додаємо. Повертаємо результат, який є з номером ij.

Тому, якщо у вас твердження справедливе для $\mathrm{k} = \mathrm{m}$, то воно буде справедливе для $\mathrm{k} = \mathrm{m}+1$, тому що там обчислюється загальна кількість маршрутів, довжини $\mathrm{m}+1$. Звідси все випливає. *Зрозуміло*?

Наслідок 1.

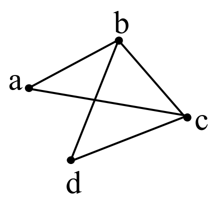
Розглянемо матрицю $\mathrm{A}^{\le \mathrm{k}} = \left\| a_{ij}^{\le k} \right\|$, де $a_{ij}^{\le k}$ – елементи цієї матриці. Ця матриця – це просто сума степенів матриці A: $\mathrm{A}^{\le \mathrm{k}} = \left\| a_{ij}^{\le k} \right\| = \mathrm{A} + \mathrm{A}^2 + A^3 + \ldots + A^k$. Тоді що в нас буде елементом матриці $\mathrm{A}^{\le \mathrm{k}}$? Тобто що це за числа такі: $a_{ij}^{\le k}$? … ☺ А як ми додаємо матриці? (*вони всі однакові* $\mathrm{m} \times \mathrm{m}$). А в результаті що? Додаємо покоординатно… і що ми тут бачимо? Елемент a_{ij}: A – кількість шляхів, довжини 1, A^2 – кількість шляхів, довжини 2, A^3 – кількість шляхів, довжини 3, …, A^k – кількість шляхів, довжини k. А загалом? Довжина всіх шляхів, довжина яких не перевищує k.

Тобто $a_{ij}^{\le k}$ = кількість шляхів, довжини $\le k$.

Наслідок 2.

Теорема залишається справедливою, якщо ми дозволяємо петлі та парні ребра. Тобто для псевдо графів та мультиграфів. Просто в цьому випадку для псевдографів ми повинні ставити 1 на діагоналі (тобто це буде шлях з вершини в вершину, довжиною 1), а для мультиграфів ставимо не 1, а кількість ребер, що поєднує дві задані вершини: якщо там 2 ребра, то ставимо 2, якщо 3 ребра, то ставимо – 3. А ось це доведення (теореми) залишається справедливим. Причому досимвольно. Це дуже зручно. *Зрозуміло*?

Давайте розглянемо певний приклад.



Візьмемо ось такий граф (рис. 5). Яка в нього буде матриця суміжності?

$$\mathrm{A} = \begin{array}{c} a \\ b \\ c \\ d \end{array} \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$ – матриця суміжності простого графа без петель

Рис. 5

Добре.

Хочу я обчислити кількість всіх маршрутів, довжини 2. Ця матриця має бути симетричною, тому, записавши перший рядок, ми можемо записати перший стовпчик. З a виходить 2 ребра, тобто в (a,a) – двійка – це власне маршрут a→b→a і a→c→a.

$$\mathrm{A}^2 = \begin{array}{c} a \\ b \\ c \\ d \end{array} \begin{bmatrix} 2 & 1 & 1 & 2 \\ 1 & 3 & 2 & 1 \\ 1 & 2 & 3 & 1 \\ 2 & 1 & 1 & 2 \end{bmatrix}.$$

Скажімо в (a,d) також повинно існувати 2 маршрути: a→b→d і a→c→d. І так само для будь-якого іншого елементу. Тобто тут кожне число одержане шляхом перебору певних шляхів з попередніх матриць. (у нас лише одна попередня матриця)

Скажімо, цікавлять мене всі маршрути, довжиною 3 з a в d: $\mathrm{a}_{\mathrm{ad}}^{(3)}$. Якщо мене цікавить лише одна пара вершин, то необов'язково обчислювати всю матрицю з a в d. Тоді я беру перший рядок у матриці A^2 і четвертий стовпчик у матриці A (*обвести їх у матрицях*) – і перемножую їх: $\mathrm{a}_{\mathrm{ad}}^{(3)} = 2$. Тобто маємо, що $\exists$ лише 2 маршрути з a в d, довжиною 3.