А **якщо** у вас **графи щільні**, тобто вони містять багато ребер, тоді списки суміжності будуть великі і незручні, – тому, що це множини. Множина – це не самий ефективний математичний об'єкт для комп'ютерної реалізації. А матриця суміжності як була $n \times n$, так і залишається $n \times n$. І працювати з нею такий самий час, як і для розріджених графів.
Щільні графи – це … *а яка взагалі в мене максимальна кількість ребер у простому графі без петель з* n *вершин?* … У вас є n *вершин, вам потрібно поєднати пари вершин ребрами… скільки можливих пар вершин у вас існує?* C_n^2 ($|E| \to C_n^2$). **Граф є щільним, коли кількість його ребер дещо завелика (близька до максимального)**.
Але знов таки. Те, що я сказала – це практичне спостереження, але це не догма. Знов таки… це залежить від того, які саме алгоритми роботи над графами ви використовуєте, які саме задачі розглядаєте.
Яка є перевага у матриці суміжності над списками суміжності?
Якщо вам потрібно перевірити чи поєднані ребром дві задані вершини, то матриця суміжності – ви просто дивитеся у відповідну комірку і все: там стоїть чи 0 або 1. Тобто ця перевірка є дуже швидкою. А в списках суміжності вам потрібно: переглянути всю цю множину – і знайти там вашу вершину. Якщо вона там є, то вони поєднані, якщо ні, то не поєднані. Так? Тобто для такої задачі матриця суміжності краща.
Але, якщо у вас стоїть задача: знайти всі вершини, з якими ваша вершина поєднана ребрами, то навпаки: список суміжності надає одразу всі такі вершини – берете їх послідовно. А матриця суміжності: треба переглядати весь рядок – і дивитися чи там є 0 чи 1. Тобто тут уже буде операція більш складною, ніж при роботі зі списками суміжності.
Приклади графів
Потрібно ввести деяку кількість графів, якими ми будемо користуватися і які ви повинні знати. І заодно покажемо, що таке граф.

1) порожній граф

Граф зветься порожнім, якщо він взагалі не має ребер. Він має спеціальне позначення: N_n і складається з множини вершин і порожньої множини ребер – $N_n = \langle V, \varnothing \rangle$. Візуально він виглядає дуже просто (рис. 1): він має 5 вершин і жодного ребра.

Рис. 1: Граф N_5

2) Повний граф – (це протилежність) це граф, який має всі можливі ребра. Він також має спеціальне позначення: $K_n = \langle V, V^{(2)} \setminus 2v \rangle$[1]. *Чому* K *– не питайте (це щось з німецької)*. А оскільки ми розглядаємо графи без петель, то всі петлі ми з множини $V^{(2)}$ (*V у 2-ому дужковому степені*) виключаємо.
Скажімо, мій улюблений граф K_5 … у вас повинно бути 5 вершин (рис. 2)

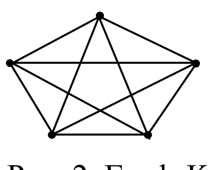


Рис. 2. Граф K_5

І всі вони повинні бути поєднані ребрами. Я би й рада намалювати цей граф, щоб у ньому ребра не перетиналися, але, як ми побачимо наприкінці нашого курсу, це не можливо. Тобто, як його не малюй, – якісь ребра все одно будуть перетинатися. Тому я вершини виділяю жирнішими.

[1] $V^{(2)}$ – це множина невпорядкованих пар, які ми трактуємо як «можливі ребра»

3) Простий ланцюг – це граф $\mathrm{P_n} = \langle V, E \rangle$,
в якому множина вершин (ми їх нумеруємо від 1 до n -ї) – $V = \{v_1 \ldots v_n\}$,
а множина ребер – (це просто пара (v_i, v_{i+1}), де i пробігає значення від 1 до n - 1) – $E = \{(v_i, v_{i+1}) \mid i = \overline{1, n-1}\}$:
Тобто, перша вершина поєднана з 2-ю, 2-га з 3-ю, 3-тя з 4-ою, 4-та з 5-ою і т.д.
Ну і візуалізація (рис. 3) – це знов таки, v_1, v_2, v_3, v_4, v_5 – і вони поєднані ребрами.

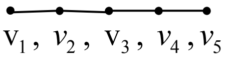


Рис. 3. Граф P_5

Чому він називається «ланцюг»? Візуально зрозуміло… ☺

4) Простий цикл – це граф, що складається з n - вершин: $C_\mathrm{n} = \langle V, E \rangle$, де $V = \{v_1 \ldots v_n\}$, $E = \{(v_i, v_{i+1}) \mid i = \overline{1, n},\ v_{n+1} \equiv v_1\}$.

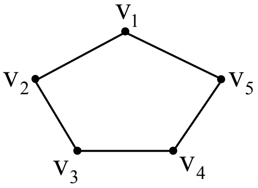


Рис. 4. Граф C_5

Простий цикл виглядає якось так: цикл, тому, що він круглий.

5) Граф, який зветься «**колесо**». Позначається: $\mathrm{W_n} = \langle V, E \rangle$. Зауважте: колесо з номером n – це граф на $n+1$-ій вершині, тобто $V = \{v_0, v_1, \ldots, v_n\}$. А ребра у нас бувають двох типів: $E = \{(v_i, v_{i+1}), (v_0, v_i) \mid i = \overline{1, n},\ v_{n+1} \equiv v_1\}$

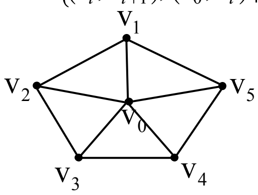


Рис. 5. Граф W_5

Маємо 6 вершин для колеса W_5. А, якщо я намалюю ребра не відрізками, а дугами – це буде, справді, колесо.
Знову ж таки… бачите, що ця картинка (рис. 5) більше зрозуміла, ніж цей запис (у п. 5), тому графи є таким потужним засобом.
Чи все вам зрозуміло?
Отже, ми визначили: що таке граф, як його можна подати.
Тепер потрібно визначити, що з цим можна робити.
По-перше, визначимо поняття «підграф».

<u>Підграф</u> $\mathrm{G}_1 \subseteq G_2$ *(будемо позначати підграф так само, як підмножини: символом «включення»; і кажемо, що граф* G_1 *є під графом графа* G_2 *тоді та тільки тоді, коли множина вершин графа* G_1 *є підмножиною множини вершин графа* G_2*, а множина ребер графа* G_1 *є підмножиною множини ребер графа* G_2*, але лише таких, які дозволяються; щоб у вас не було ситуації, коли у вас вершини a та b і ребро* (b, c)*, що ось за таким визначенням:* $\mathrm{E}_1 \subseteq E_2$ *буде коректним, але це не буде граф (тобто у мене є* E_1 *– ребра* *можуть бути лише з цієї множини –* $V_1^{(2)}$ *– і, якщо це буде підмножина* E_2*– це буде підграф))* $\Leftrightarrow (V_1 \subseteq V_2) \& (E_1 \subseteq E_2 \cap V_1^{(2)})$.

І, відповідно, поняття «підграфа» дозволяє нам ввести певний «**універсальний граф**», в якому ми будемо виконувати алгебраїчні операції. І таким графом, зазвичай, є **повний граф** K_n**, тобто граф, у якому є всі вершини і всі ребра, які ми можемо побудувати**.

Ще для нас цікаві так звані «кістякові підграфи».

Кістяковий підграф (*рос.* остовной («остов» – каркас) підграф; *англ.* spanning *(накриваючий)* subgraph; *укр. може бути «кістяковий», «каркасний», ... (пів десятка назв)*) – це підграф, який містить всі вершини нашого графу G_2, але не всі ребра.

Ми кажемо, що G_1 **є кістяковим підграфом** G_2**, якщо** $V_1 = V_2$ (множини вершин у них співпадають), **і, відповідно,** $E_1 \subseteq E_2$ (множина ребер E_1 є підмножиною ребер E_2).

Тепер

Алгебраїчні операції над графами

Оскільки граф – це пара множин, то логічно, що я можу операції над множинами поширити на графи. Саме це я зараз і зроблю.

1) Об'єднання графів.

Скажемо, що граф G є об'єднанням графів G_1 та G_2, якщо множини вершин V – це буде об'єднанням вершин V_1 *та* V_2**, а множина ребер** E **– це буде об'єднанням множин ребер** E_1 *та* E_2:

$G = G_1 \cup G_2 \Leftrightarrow V = V_1 \cup V_2, E = E_1 \cup E_2$**.**

Тут, здається, все зрозуміло…

2) Перетин графів.

$G = G_1 \cap G_2 \Leftrightarrow \begin{cases} V = V_1 \cap V_2, \\ E = E_1 \cap E_2 \cap V^{(2)} \end{cases}$ *($\cap V^{(2)}$ – щоб не вийти за межі визначення графа, щоб у нас не з'явилися у перетині ребра, яких не має у множині пар першин).*

3) Різниця графів.

Існує два способи визначити різницю графів. І, коли ви будете читати джерела, ви повинні обов'язково дивитися що саме під терміном «різниця графів» розуміється.

<u>По-перше</u>, ми кажемо, що граф G є різницею графів G_1 та G_2:

$G = G_1 \setminus G_2 \Leftrightarrow \begin{cases} V = V_1 \setminus V_2, \\ E = (E_1 \setminus E_2) \cap V^{(2)} \end{cases}$ **(**$E = (E_1 \setminus E_2) \cap V^{(2)}$ *– множина ребер є різницею множин ребер* E_1 *та* E_1 *в межах допустимого***).**

Тобто в цьому визначенні під різницею ми розуміємо: взяти граф G_1 і викинути з нього все, що пов'язане з графом G_2: всі вершини, всі ребра, всі ребра, що зв'язували вершини.

<u>Друге визначення різниці.</u>

Якщо у вас графи G_1 та G_2 визначені на одній множині вершин ($V_1 = V_2$), то різниця графів – це буде просто різниця множини ребер:

$$G = G_1 \setminus G_2 \Leftrightarrow \begin{cases} V = V_1, \\ E = E_1 \setminus E_2 \end{cases} .$$

Я ще раз наголошую: це не одне і те ж саме визначення. За альтернативою – це два різних визначення фактично двох різних операцій.

Самі подивіться, якщо в 1) – визначення підставити $V_1 = V_2$, то G буде порожньою множиною (оскільки в нього множина вершин порожня). А в 2) – ні. І іноді потрібно використовувати таку – 1) операцію, іноді – таку: 2). Потрібно дивитися, що саме під «різницею графів» розуміє автор.

З 2-го визначення різниці графів випливає означення «доповнення до графа».

4) Доповнення до графа.

Кажемо, що $\overline{\mathrm{G}}$ є доповненням до графа G: $\overline{\mathrm{G}} = \left\langle \mathrm{V}, \mathrm{V}^{(2)} \setminus \left(E \cup vv\right)\right\rangle$, (тобто це та сама множина вершин V, окрім тих, що входять у множину G).

Звісно, що доповнення – це буде різницею повного графа $\mathrm{K_n}$ та G (через другий спосіб).

Наступна операція, яка зветься «сума графів» або «сполучення графів» або «поєднання графів».

5) Сума графів.

$\mathrm{G} = \mathrm{G}_1 + G_2 \Leftrightarrow \begin{cases} V = V_1 \coprod V_2 \\ E = E_1 \cup E_2 \cup V_1 \times V_2 \end{cases}$ ($V = V_1 \coprod V_2$ – сума визначена для графів, які не мають спільних вершин (тобто об'єднання цих двох множин повинне бути диз'юнктним – вони не повинні мати спільних точок. Чому? За визначенням!), *бо якщо вони мають спільні вершини, то все йде погано*); (множини ребер об'єднуються і додаються всі можливі ребра між вершинами V_1 та V_2 – зараз я їх записала, як декартовий добуток $V_1 \times V_2$, але ви пам'ятаєте, що тут невпорядковані пари, тобто це всі можливі невпорядковані пари, в яких одна пара з множини V_1, а друга – з множини V_2)[2].

Скажімо, беру я граф $\mathrm{K_1}$. Що таке граф $\mathrm{K_1}$? Це точка. Повний граф з однієї вершини. Вершина одна, а ребер – жодного.

Беру я граф C_5. Це, як ми пам'ятаємо, ось такий цикл:

Що буде, якщо я додам граф $\mathrm{K_1}$ до графа C_5:

$\mathrm{K_1} + C_5 = \mathrm{W_5}$

Значить, я повинна взяти граф C_5 повністю (рис. 6), я повинна взяти граф $\mathrm{K_1}$ повністю, а потім поєднати всі вершини графа $\mathrm{K_1}$ з усіма вершинами графа C_5. Що я маю? Колесо (рис. 7).

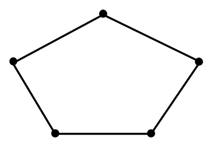


Рис. 6. Граф C_5

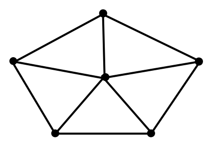
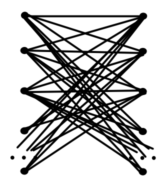


Рис. 7. Граф W_5 колесо

Візьмемо порожній граф $\mathrm{N_n}$ і порожній граф $\mathrm{N_m}$. Що буде, якщо їх поєднати? Рис. 8



$\mathrm{N_n}$ $\mathrm{N_m}$

Рис. 8. Граф $\mathrm{K_{n,m}} = \mathrm{N_n} + \mathrm{N_m}$

[2] Підкреслене записати словами

Давайте для прикладу: $N_3 + N_3$? Маємо таку картинку: рис. 9. (перший граф N_3 – це у вас 3 будинки, другий граф N_3 – це у вас 3 криниці, а граф $N_3 + N_3$ – це всі можливі шляхи між будинками та криницями). Зветься він «повний двочастковий граф», але ми це ще введемо. Позначається – $K_{3,3}$. А в попередньому випадку (рис. 8) – $K_{n,m}$.

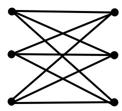


Рис. 9. Граф $K_{3,3} = N_3 + N_3$

І знов таки, я і рада намалювати, щоб ребра не перетиналися, але неможливо. Зрозуміло?
Так ось… що буде, якщо від цієї суми ($K_{3,3} = N_3 + N_3$) відняти N_3?

Якщо розглядати різницю у другому сенсі, то нічого гарного не вийде. А якщо відняти у першому сенсі, то ми видаляємо всі вершини графа N_3 (другий доданок) і всі ребра, що його поєднують – одержимо перший доданок – N_3.

Тобто, <u>якщо нам потрібно маніпулювати сумами, то тут різницю краще брати в першому сенсі</u>.

<u>Якщо вам потрібно маніпулювати з доповненнями, то тут різницю треба брати в другому сенсі.</u>

В грубому смислі: $(G_1 + G_2) \setminus G_2 = G_1$ – в сенсі 3а). Але користуйтесь цим з обережністю.

Так…Там ще є декартовий добуток, доречі, але я його зараз не буду вводити. *Не треба такої реакції… ☺я не буду його вводити взагалі… ☺*

Мені буде простіше його ввести при практичній реалізації теорії автоматів… якщо буде час, то ми його там і введемо.

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…але це на наступній парі…