

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ”

Кафедра математики факультету інформатики

**Курсова робота на тему:
Графи перетинів**

Керівник курсової роботи:
к. ф.-м. н. *Козеренко С.О.*
(*прізвище та ініціали*)

(*підпис*)
“ _____ ” _____ 2021 р.

Виконав студент
3-го року навчання спеціальності
121 “Інженерія програмного забезпечення”
Гак Артем Олегович
(*ПІБ*)

Київ – 2021

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ”

Кафедра математики факультету інформатики

ЗАТВЕРДЖУЮ
Зав. кафедри математики,
проф. д,ф-м.н.
_____ Б.В. Олійник
(підпис)
“ _____ ” _____ 2021 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на курсову роботу
студенту 3-го курсу факультету інформатики
Гаку Артему Олеговичу

Тема: Графи перетинів.

Вихідні дані: Досліджено властивості графів перетинів.

Зміст ТЧ до курсової роботи:

- Індивідуальне завдання
- Анотація
- Вступ
- 1 Основні означення
- 2 Основні результати
- 2.1 Число перетинів
- 2.2 Графи блоків
- 2.3 Реберні графи
- 2.3 Графи клік
- 2.3 Хордальні графи
- Висновки
- Література

Дата видачі “ _____ ” _____ 2021 р. Керівник _____

(підпис)

Завдання отримав _____

(підпис)

Тема: Графи перетинів.

Календарний план виконання роботи:

Номер	Назва етапу курсової	Термін виконання етапу	Примітка
1.	Отримання теми курсової роботи.	вересень	
2.	Ознайомлення з темою курсової.	вересень	
3.	Розробка плану та структури роботи.	жовтень	
4.	Робота з науковою літературою, опис основних означень теорії графів.	листопад	
5.	Дослідження основних властивостей графів перетинів	листопад-грудень	
6.	Дослідження класів графів перетинів. Одержання основних результатів	січень-березень	
7.	Робота над текстовим оформленням результатів.	квітень	
8.	Попередній аналіз курсової. Виправлення помилок.	початок травня	
9.	Захист курсової роботи. роботи.	середина травня	

Зміст

Анотація	5
Вступ	6
1 Основні означення	7
2 Основні результати	9
2.1 Число перетинів	9
2.2 Графи блоків	12
2.2.1 Означення	12
2.2.2 Характеризація	14
2.2.3 Ітерації блокового оператора	16
2.2.4 Опис алгоритмів із доданого коду	20
2.3 Реберні граfi	22
2.3.1 Означення	22
2.3.2 Характеризація	26
2.4 Граfi клік	32
2.4.1 Означення	32
2.4.2 Оператор клік, характеризування	35
2.5 Хордальні граfi	36
2.5.1 Означення	36
2.5.2 Характеризація	38
Висновки	39
Література	40

Анотація

Нехай X деяка множина, а $\mathcal{F} \subset 2^X$ родина її підмножин. *Графом перетинів* $\Omega(X, \mathcal{F})$ називається граф Ω з множиною вершин $V(\Omega) = \mathcal{F}$ та дві його вершини A, B вважаються суміжними, якщо $A \cap B \neq \emptyset$.

Мета роботи полягає у дослідженні графів перетинів, зокрема чотирьох класів: графів блоків, реберних графів, графів клік та хордальних графів.

Ключові слова: графи перетинів, блок, графи блоків, оператор блоку, реберні графи, кліка, графи клік, хорда, хордальні графи.

Вступ

Графи перетину це досить перспективний напрямок теорії графів. Вивченням даного питання займалися Оре, Сабідуссі, Байнеке, Кастейлен та інші. Теорія графів має широке застосування у біології, комп'ютерних науках, матричному аналізі та статистиці[8].

Ми маємо справу з різносторонньою, цікавою темою, при дослідженні якої, одержимо дещо інший погляд на всю теорію графів.

Вершини графа є множинами і вершини є суміжними, якщо їх множини мають непорожній перетин.

Варто зазначити, що кожен граф можна подати як граф перетинів, відповідний алгоритм представлений у роботі.

Оскільки це поняття є досить широке, тому природно звужувати на окремі класи. В цій роботі розглядаються 4 класи: реберні графи, графи блоків, графи клік та хордальні графи. Для кожного з вищеповисаних класів надаються відповідні характеристики.

Реберні графи найбільш вивчені. Важливу роль грає теорема Уїтні. Вона показує, що граф повністю визначається своїм реберним графом, крім одного випадку[3]. У своїй роботі Байнеке [1] дає характеристику в термінах заборонених підграфів. Ми ж дамо в термінах розбиття на повні підграфи.

Графи блоків – це графи у яких кожна компонента двозв'язності є клікою. Для цього класу ми даємо характеристику, як граф, кожен блок, якого є повним підграфом. Далі ми розглянемо ітерації блокового оператора, і дамо оцінки на необхідну кількість застосувань блокового оператора для зведення графа до одноточкового, ланцюга чи дерева.

Кліки є однією з базових концепцій теорії графів. Вони часто виникають в задачах інформатики.

Хордальні графи – це графи, що не містять породжених циклів довжини більшої за 3. Кожен граф можна представити як граф перетинів підграфів деякого графа. Хордальні ж характеризуються як граф перетинів піддерев у дереві.

Мета роботи полягає у дослідженні вищенаведених класів графів перетинів. Для кожного з них будуть надані характеристики, деякі результати ілюструватимуться малюнками. На відміну від ітерацій реберного графа, блоковий оператор не був досліджений раніше. Саме у цьому полягає новизна цієї роботи. Ми розглянемо кількості ітерацій, необхідні, щоб звести граф до дерева, ланцюга та одноточкового графа.

1 Основні означення

Означення 1.1. Неорієнтований граф G — це впорядкована пара (V, E) , де V — множина вершин або вузлів, E — множина пар (у випадку неорієнтованого графу — невпорядкованих) вершин з V , які називаються ребрами.

Означення 1.2. Пара вершин графа i, j графа G називається *суміжними*, якщо $ij \in E(G)$.

Означення 1.3. Графи G і H називаються *ізоморфними*, якщо існує бієкція між їх множинами вершин $f : V(G) \rightarrow V(H)$ така, що будь-які дві вершини u і v графа G суміжні в G тоді і тільки тоді, коли u і v суміжні в H . Позначається $G \simeq H$.

Означення 1.4. Граф H є *породженим підграфом* графа G , якщо $V(H) \subseteq V(G)$ та $\forall u, v \in V(H) : uv \in E(H) \Leftrightarrow uv \in E(G)$.

Означення 1.5. Граф називається *зв'язним*, якщо між кожною парою його вершин існує шлях, який їх сполучає.

Означення 1.6. *Компонентою зв'язності* графа називається його максимальний (за включенням) зв'язний підграф.

Означення 1.7. *Уніциклічний граф* — граф з єдиним циклом.

Означення 1.8. Вершина $u \in V(G)$ графа G називається *точкою з'єднання*, якщо її видалення збільшує кількість компонент зв'язності.

Означення 1.9. *Околом* $N(a)$ вершини a графа G називається множина вершин, суміжних із нею, тобто $N(a) = \{u \mid u \in V(G), ua \in E(G)\}$.

Означення 1.10. Граф називається *двозв'язним*, якщо він не містить точок з'єднання.

Означення 1.11. *Блоком графа* G називається його максимальний (за включенням) двозв'язний підграф.

Означення 1.12. *Повний граф* — той у якого, кожна пара вершин з'єднана ребром, позначається K_n , де n — кількість вершин.

Означення 1.13. Нехай X деяка множина, а $\mathcal{F} \subset 2^X$ родина її підмножин. *Графом перетинів* $\Omega(X, \mathcal{F})$ називається граф Ω з множиною вершин $V(\Omega) = \mathcal{F}$ та дві його вершини A, B вважаються суміжними, якщо $A \cap B \neq \emptyset$.

Приклад 1.14. Граф перетинів $G \simeq \Omega(X, \mathcal{F})$, де $X = \{a, b, c, d, e, f, g, h\}$,
 $\mathcal{F} = \{\{a\}, \{a\}, \{a, b\}, \{b, c, f\}, \{f, g, h\}, \{h\}, \{f, g, h\}, \{g\}, \{c, d, e\}, \{d\}, \{e\}\}$.

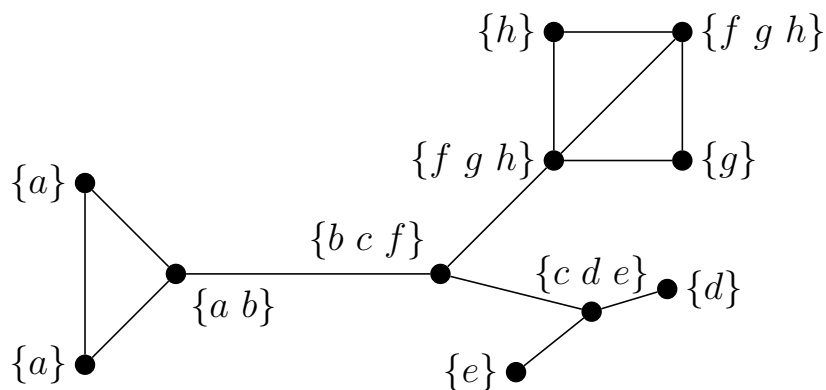


Рис. 1.1. Граф перетинів

2 Основні результати

2.1 Число перетинів

Твердження 2.1. *Кожен граф ізоморфний деякому графу перетинів.*

Доведення. Нехай G деякий граф, побудуємо ізоморфний йому граф перетинів Ω таким чином:

Позначимо $M(v)$ – множину графа перетинів Ω , що відповідає вершині v графа G . Тоді $\forall v_i \in V(G) : M(v_i) = \{v_i\} \cup E(v_i)$, де $E(v_i)$ – множина ребер, інцидентних вершині v_i . Таким чином отримаємо граф перетинів $\Omega(X, \mathcal{F})$, де $X = V(G) \cup E(G)$, інакше кажучи $X = \{v_1, v_2, \dots, e_1, e_2, \dots\}$, а $\mathcal{F} = \{M(v) : v \in V(G)\}$.

Зауваження:

1. $\mathcal{F} \subset 2^X$ за побудовою.
2. Граф Ω отриманий в ході такої побудови є ізоморфний даному початковому графу G , оскільки:

2.1. Кожна ізольована вершина в G є ізольованою в Ω , оскільки множина кожної такої вершини має один унікальний елемент: $\forall n : deg(v_n) = 0 : M(v_n) = \{v_n\}$

2.2. $\forall v_G, w_G \in V(G), w \neq v : v_G$ та w_G – суміжні $\Rightarrow v_\Omega$ та w_Ω – суміжні. Оскільки v_G інцидентна w_G , то $\exists e_k$ – ребро, що сполучає ці вершини. За побудовою це ребро e_k є у множинах вершин v_Ω, w_Ω , таким чином $M(w_\Omega) \cap M(v_\Omega) = e_k$, тому вершини v_Ω та w_Ω є інцидентними.

2.3. $\forall v_G, w_G \in V(G), w \neq v : v_G$ та w_G не суміжні $\Rightarrow v_\Omega$ та w_Ω не суміжні. Розглянемо множину ребер інцидентних v_G , оскільки вершини v_G та w_G не суміжні, то жодне ребро з цієї множини не інцидентне w_G , тому $\forall e \in M(v_\Omega) : e \notin M(w_\Omega)$. Аналогічно $\forall e \in M(w_\Omega) : e \notin M(v_\Omega)$. Тому $M(v_\Omega) \cap M(w_\Omega) = \emptyset$, тому вершини v_Ω та w_Ω не суміжні. □

Означення 2.2. *Числом перетинів скінченного графа G називається*

$$\omega(G) = \min\{|X| : \Omega(X, \mathcal{F}) \simeq G \text{ для деякої родини } \mathcal{F} \subset 2^X\}.$$

Надалі розглядаються лише скінченні графи.

Твердження 2.3. *Для зв'язного графа G із ≥ 3 вершинами має місце нерівність $\omega(G) \leq |E(G)|$.*

Доведення. Оскільки маємо зв'язний граф з $|V(G)| \geq 3$, то його можна представити як у твердженні 3.1, але надавши кожній вершині лише множину інцидентних їй ребер. Додавання самого елемента v_k до вершини непотрібне, бо граф зв'язний, і не буде проблем із компонентами виду K_1 та K_2 . Таким чином $|X| \leq |E(G)|$, а оскільки $\omega(G)$ – мінімальне значення $|X|$, то $\omega(G) \leq |E(G)|$. (Графи будуть ізоморфними за 2.2 та 2.3) □

Означення 2.4. Кажуть, що граф G не містить трикутників, якщо повний граф K_3 не є підграфом G .

Теорема 2.5. Для зв'язного графа G із ≥ 4 вершинами має місце рівність $\omega(G) = |E(G)|$ тоді й тільки тоді, коли G не містить трикутників.

Доведення. \Rightarrow Якщо $\omega(G) = |E(G)|$, то G не містить K_3

Від супротивного, припустимо, що це не так, тоді існує граф G , для якого виконується рівність $\omega(G) = |E(G)|$, і який містить K_3 . Розглянемо локальне місце в графі з трикутником:

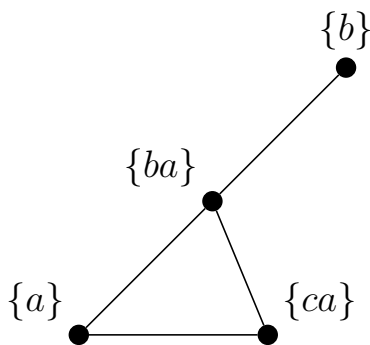


Рис. 2.1. Підграф із трикутником

Оскільки граф зв'язний то принаймні одна вершина із K_3 буде з'єднана ребром з якоюсь іншою вершиною (не з K_3). Зазначимо, що для того, щоб представити цю частину графа, достатньо множини $X = \{a, b, c\}$ з 3 елементів (рис. 1). Таким чином ми представили 4 ребра за допомогою 3 множини з 3 елементів. Щоб представити решту $|E(G)| - 4$ ребер графа алгоритмом з Твердження 3.1 знадобиться $|E(G)| - 4$ елементів множини. Отже, для всього графа нам знадобиться $|E(G)| - 1$ елемент, що суперечить припущенню $\omega(G) = |E(G)|$. Тобто виконується нерівність $\omega(G) \leq |E(G)|$.

\Leftarrow Нехай маємо G граф, та $\Omega(X, \mathcal{F})$ його граф перетинів. Оскільки G не містить трикутників, $\forall a \in X : a$ може лежати у не більше 2 множинах із \mathcal{F} . $\forall e \in E(G) : F_u \cap F_v \neq \emptyset$, де F_u та F_v - множини вершин u та v , які сполучаються ребром e . $|F_u \cap F_v| = 1$, якби це число було б більшим за 1, то можна було б видалити всі спільні елементи з множини $|F_u \cap F_v|$ крім одного, що суперечило б мінімальності в означенні $\omega(G)$. Нехай $|F_u \cap F_v| = \{x_e\}$, де e - ребро, що сполучає вершини v та u . Таким чином можна поставити у відповідність $e \rightarrow x_e$, а отже існує ін'єкція : $E(G) \rightarrow X$. Що доводить нерівність $\omega(G) \geq |E(G)|$

□

Твердження 2.6. Для повного графа K_n , $n \geq 4$ виконується рівність

$$\omega(G) = 1 + \lceil \log_2 n \rceil$$

Доведення. Нехай G – повний граф та $V = \{v_1, v_2, v_3, \dots\}$ – множина його вершин. Потрібно знайти $\min N = |X|, \Omega(X, \mathcal{F})$.

Використаємо теорему Ердеша[2], яка говорить, що 2^{n-1} розмір найбільшої сім'ї підмножин перетинів на булеані 2^n . Таким чином можна встановити бієкцію між цими проблемами. Множини вершин графа – це множини сім'ї перетинів, а булеан – це $2^{|X|}$.

Щоб знайти $\omega(K_n)$ потрібно мінімізувати число $|X|$ при цьому забезпечивши нерівність $2^{|X|-1} \geq n$.

Розв'яжемо відносно $|X|$:

$$|X| - 1 \geq \log_2 n$$

$$|X| \geq \log_2 n + 1$$

Оскільки ми шукаємо найменше ціле значення:

$$|X| = 1 + \lceil \log_2 n \rceil$$

□

2.2 Графи блоків

2.2.1 Означення

Означення 2.7. Графом блоків $B(G)$ графа G називається граф перетинів $\Omega(V(G), \mathcal{B})$, де \mathcal{B} – це родина множин вершин блоків в G . Зауважимо, що два блоки B_1, B_2 можуть перетинатися лише по одній вершині (яка є точкою з'єднання графа G).

Наприклад, нехай граф G має такий вигляд:

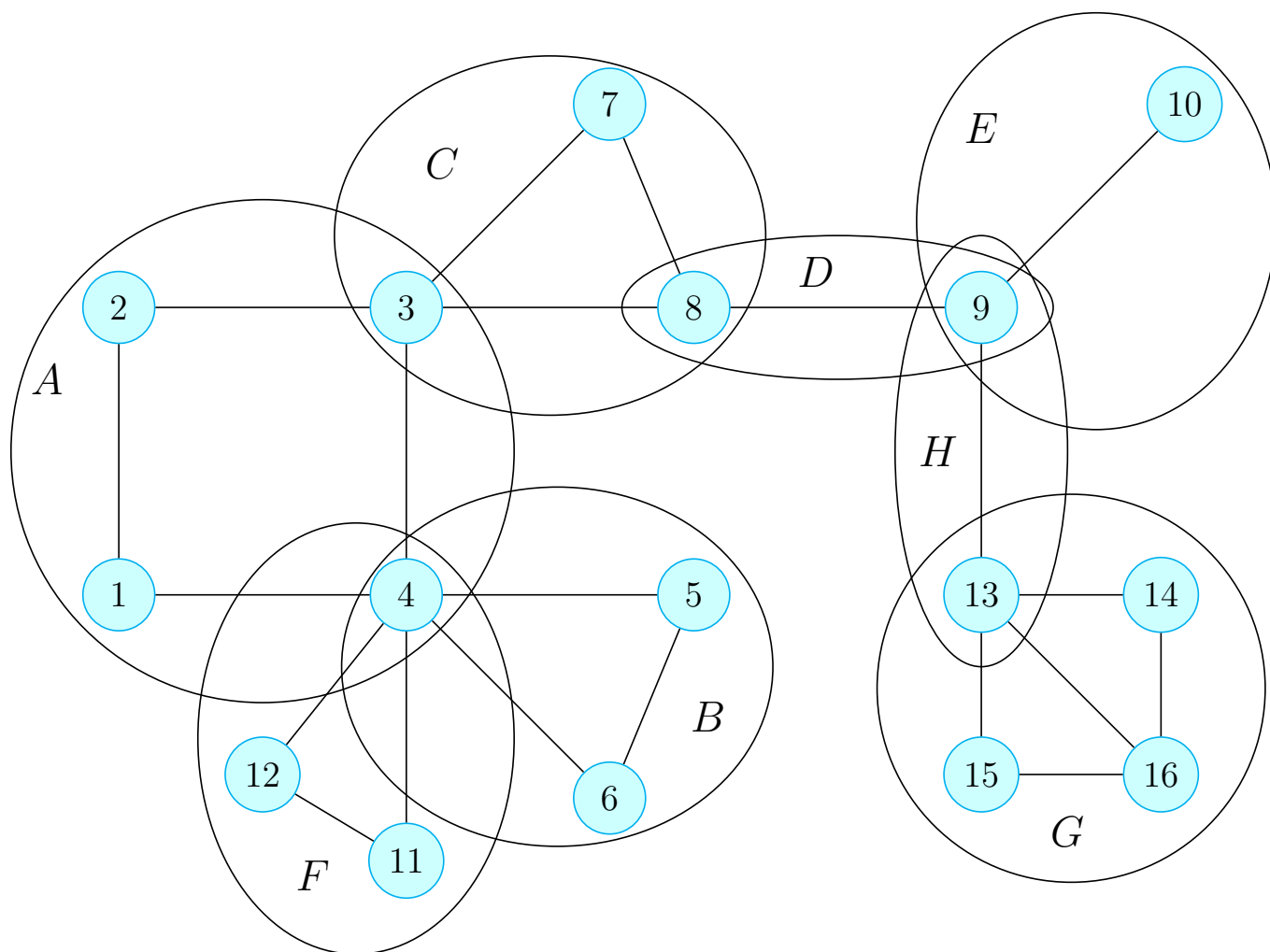
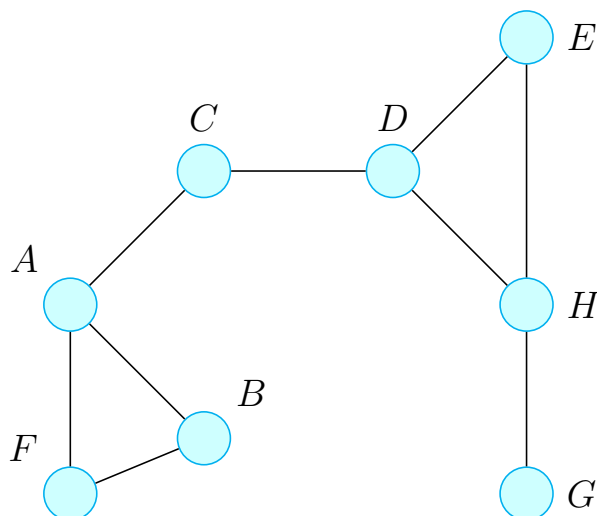


Рис. 2.2. Граф G

Тоді його граф блоків $B(G)$ буде таким:

Рис. 2.3. Граф блоків $B(G)$

Лема 2.8. Зв'язний граф G (із $|V(G)| \geq 3$) є двозв'язним тоді і тільки тоді, коли будь-які дві його вершини лежать на деякому спільному простому циклі.

Доведення. \Leftarrow Нехай G граф, такий, що будь-які 2 вершини лежать на спільному простому циклі. Від супротивного, припустимо, що G не є блоком, тоді існує точка зв'язності p , при видаленні якої утвориться 2 компоненти зв'язності. Тоді існують мінімум 2 вершини між якими не буде шляху при видаленні p . Зафіксуємо ці вершини як u та v . Оскільки за умовою вони лежать на одному простому циклі, нехай

$P = \{u - u_0 - u_1 - \dots - u_n - v - v_0 - v_1 - v_2 - \dots - v_m - u\}$. То існує мінімум два різних простих шляхи між u та v , зафіксуємо їх.

Нехай $Q_1 = \{u - u_0 - u_1 - \dots - u_n - v\}$ та $Q_2 = \{v - v_0 - v_1 - \dots - v_m - u\}$. $\forall i : v_i \neq u_i$, оскільки P – простий цикл. Щоб прибрати зв'язок між вершинами u та v потрібно принаймні розірвати 2 ланцюги Q_1 та Q_2 , а оскільки вони мають різні вершини, то це не можливо досягти видаливши лише одну вершину p , що суперечить припущенню.

\Rightarrow Якщо граф двозв'язний, тоді будь-які дві різні вершини лежать на деякому простому циклі. Нехай G – двозв'язний граф, а u та v – дві його вершини, $u \neq v$. Позначимо U – множина вершин відмінних від u , які лежать на простому циклі, що містить u . Оскільки $|V(G)| \geq 3$, а також, G не має точок зв'язності, то G не

містить мостів. Тому кожна вершина, що суміжна u міститься в U , і $U \neq \emptyset$.



Рис.

2.4. Ілюстрація 1 до Лема 2.8

Припустимо, що v не лежить в U . Нехай w – вершина в U , така, що відстань $d(w, v)$ мінімальна. Нехай P_0 – найкоротший простий $(u-v)$ -ланцюг, а P_1 і P_2 – два простих $(u-v)$ -ланцюги циклу, що містить u та w (Рис. 4).

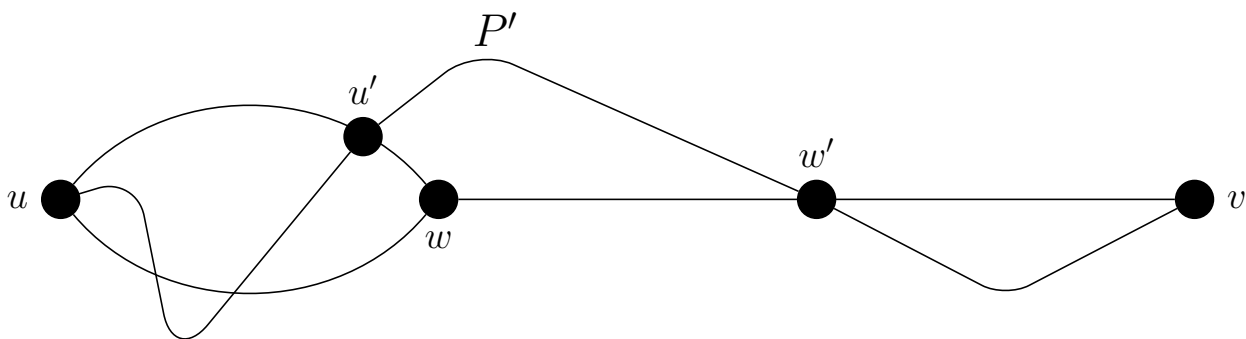


Рис. 2.5. Ілюстрація 2 до Лема 2.8

Оскільки w не є точкою зв'язності, то існує простий $(u-v)$ -ланцюг P' , який не містить w (Рис. 5). Позначимо w' найближчу вершину до u , що належить P' , яка водночас належить і P_0 . Через u' позначимо останню вершину $(u-w')$ -ланцюга в P' , яка належить або P_1 або P_2 . Не втрачаючи загальності, припустимо, що u' належить P_1 .

Нехай Q_1 – простий $(u-w')$ -ланцюг, що повністю вміщає $(u-u')$ -підланцюг ланцюга P_1 та $(u'-w')$ -підланцюг ланцюга P' . А Q_2 – простий $(u-w')$ -ланцюг, що вміщає P_2 та $(w-w')$ -підланцюг ланцюга P_0 . За побудовою Q_1 та Q_2 – два простих $(u-w')$ -ланцюга, що не перетинаються. Разом вони утворюють простий цикл, тому w' належить U . Оскільки w' належить найкоротшому ланцюгу, $d(w', v) < d(w, v)$. Це суперечить вибору w та доводить, що u та v належать одному простому циклові. \square

2.2.2 Характеризація

Означення 2.9. Точки зв'язності будемо називати *червоними* вершинами, а інші вершини – *зеленими*.

Теорема 2.10. Для графа H існує граф G із $B(G) \simeq H$ тоді й тільки тоді, коли кожен блок в H є повним графом.

Доведення. \Rightarrow Якщо існує граф $G : B(G) \simeq H$, то кожен блок в H є повним графом. Нехай $H = B(G)$. Від супротивного, припустимо, що в H є блок P , який не є повним графом. Тоді в P знайдуться дві не суміжні вершини u та v . За лемою 3.5 вони належать одному спільному циклу Z . Нехай V_Z множина вершин з блоку P , що належать циклу Z . Тоді об'єднання блоків графа G , що відповідають вершинам з V_Z є зв'язним графом, без точок зв'язності (тому, що Z – цикл). Оскільки це об'єднання не містить точок зв'язності, то це суперечить максимальності в означенні блоку графа. Тобто P не є максимальним підграфом без точок зв'язності, щоб вважатися блоком.

\Leftarrow Якщо кожен блок в H є повним графом, то існує граф $G : B(G) \simeq H$. Наведемо алгоритм побудови графа G для H . Нехай маємо граф H у якому кожен блок є повним графом.

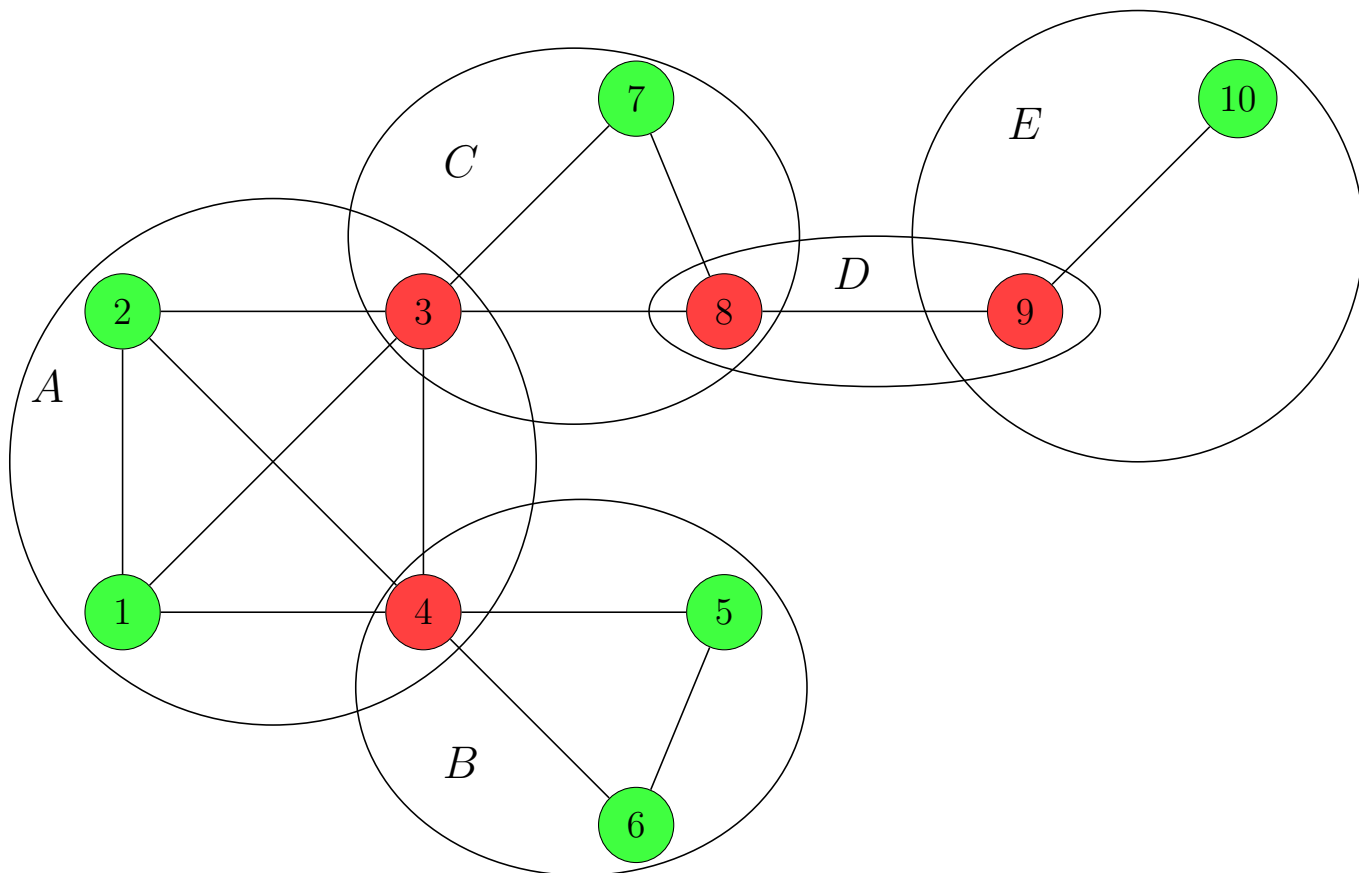
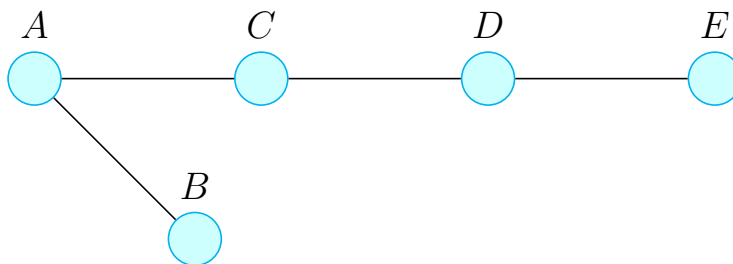
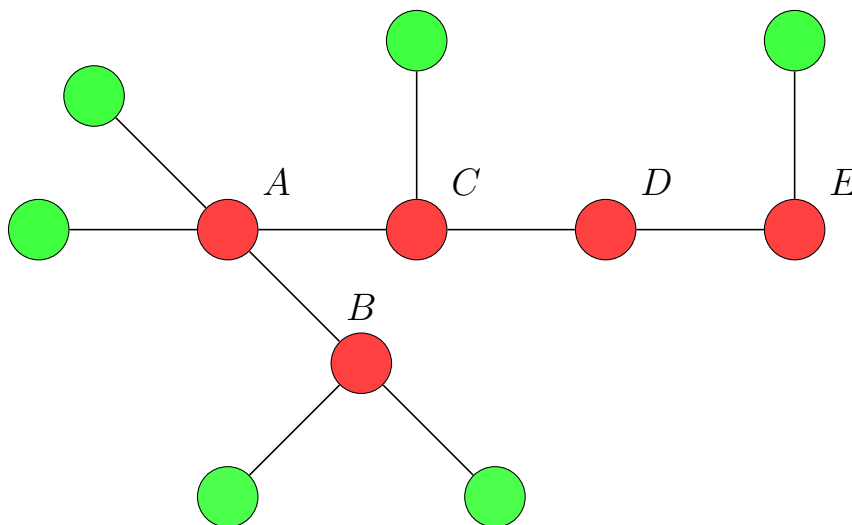


Рис. 2.6. Граф H

Побудуємо граф $B(H)$:

Рис. 2.7. Граф $B(H)$

Додамо до кожної вершини отриманого графа таку кількість висячих, як у відповідному їй блоці графа H .

Рис. 2.8 . Граф G

Отриманий граф G , буде таким, що $B(G) \simeq H$. □

2.2.3 Ітерації блокового оператора

Означення 2.11. Для числа $n \in \mathbb{Z}_+$ через $B^n(G)$ будемо позначати n -ту ітерацію оператора B на графі G (при цьому, звичайно $B^0(G) = G$). Легко бачити, що для зв'язного графа G із $|V(G)| \geq 2$ має місце нерівність $|V(B(G))| < |V(G)|$. Таким чином, для деякого $n \in \mathbb{N}$ виконано $B^n(G) \simeq K_1$. Покладемо $\beta_0(G) = \min\{n \in \mathbb{Z}_n : B^n(G) \simeq K_1\}$ для кожного зв'язного графа G .

Теорема 2.12. Для зв'язного графа блоків G має місце рівність

$$\beta_0(G) = \text{diam } G.$$

Доведення. Основний крок: показати рівність $\text{diam } B(G) = \text{diam } G - 1$ для всіх графів блоків G із $|V(G)| \geq 2$ (далі індукція по $\text{diam } G$).

Покажемо, що ланцюг, що відповідає діаметру графа має вигляд $\{g - r - r - \dots - r - r - g\}$, де g – зелена, r – червона вершина.

Чому це так?

1) З максимальності в означення діаметра графа випливає, що такий ланцюг буде проходити по найбільшій кількості блоків. Таким чином, щоб пересуватись між блоками, у цьому ланцюгу мають бути червоні вершини. (Якби їх не було, то ми б не могли побудувати шлях між двома зеленими вершинами різних блоків).

2) В середині ланцюга не може бути зелених вершин. від супротивного, нехай це можливо: $\{g - r - \dots - r - g - r - \dots - r - g\}$. розглянемо частину шляху, де всередині зустрічається зелена вершина. Тоді є 2 варіанти:

Індекси позначають блоки, яким належить ця вершина. Зелені можуть належати лише одному блоку, червоні декільком, бо є точками зв'язності. Ми розглядаємо лише блоки залучені в нашому ланцюгу-діаметрі, тому якщо червона належить іншим блокам, що не беруть участі у вибраному ланцюгу, ми їх не пишемо в індекси.

$$\dots - r_{A,B} - g_B - r_{B,C} - \dots$$

Даний варіант суперечить властивості мінімального шляху в означенні діаметра, бо оскільки права і ліва червоні вершини обоє належать блоку B то можна дістатись з однієї в іншу без проходження по вершині g . (Оскільки кожен блок графа G є повним).

$$\dots - r_{A,B} - g_B - r_{A,B} - \dots$$

Даний варіант теж не підходить з міркувань наведених вище. А також має іншу суперечності, або ліва і права вершини r – це одна й та ж вершина, то у шляху вершини повторюються, що суперечить мінімальності відстані, а якщо ні, то блоки A та B перетинаються по більш ніж одній точці зв'язності, що суперечить максимальності в означенні блоку.

3) На обох кінцях такого ланцюга є по одній зеленій вершині. Це так, бо діаметр – це найбільший шлях. Якби на кінцях не було зелених вершин. То можна було зробити крок на яку-небудь зелену вершину із крайніх блоків, і отримати більший діаметр, ніж початковий.

Ми показали, що ланцюг діаметра має саме такий вигляд:

$$\{g - r - r - \dots - r - r - g\}$$

Покажемо, що застосування оператора блоків скорочує будь-який ланцюг такого виду на одну вершину. Позначимо індексами блоки, яким належать вершини:

$$L = \{g_0 - r_{0,1} - r_{1,2} - \dots - r_{n-2,n-1} - r_{n-1,n} - g_n\}$$

Нехай $H = B(G)$ При застосуванні оператора блоків, в отриманому графі кожна вершина відповідає якомусь блоку з початкового, розглянемо вершини $0, 1, \dots, n - 1, n$ у графі H , що відповідають блокам у графі G . Вони всі будуть з'єднані одним ланцюгом, наступним чином:

$K = \{0 - 1 - 2 - \dots - (n - 2) - (n - 1) - n\}$. Ланцюг K має на одну вершину менше, ніж ланцюг L . Таким чином при кожній ітерації оператора блоків, ланцюг діаметра зменшується на 1. Отже, на $diam(G)$ застосувань діаметр буде 0, а отже

матимемо граф K_1 .

В загальному випадку, для довільного графа G : $n = \text{diam}(B(G)) + 1$. За перше застосування згідно з твердженням 3.6 отримуємо граф $B(G)$, кожен блок якого є повним. А за наступні $\text{diam}(B(G))$, граф $B(G)$ перетвориться в K_1 за вище доведеним. □

Наслідок 2.13. Для зв'язного графа G з $|V(G)| \geq 2$ виконується

$$\beta_0(G) = \text{diam } B(G) + 1.$$

Доведення. За Теоремою 2.12, $\beta_0(G) = \beta_0(B(G)) + 1 = \text{diam } B(G) + 1$. □

Твердження 2.14. $B(G)$ є ланцюгом тоді й тільки тоді, коли кожна червона вершина G – лежить рівно у 2 блоках та кожен блок містить $n \leq 2$ червоні вершини.

Доведення. \Leftarrow Нехай маємо граф G як в умові. Оскільки кожна червона лежить рівно у 2 блоках, то при застосуванні блокового оператора, кожна червона дасть 1 ребро, що з'єднає два блоки в яких вона лежить. А скільки кожен блок має $n \leq 2$ червоні вершини, відповідна йому вершина у графі $B(G)$ матиме степінь $n \leq 2$ тому, що кожна червона дає рівно одне ребро. Тому граф $B(G)$ містить лише вершини степеня $n \leq 2$. Простим циклом C_k , $k \geq 4$ він бути не може, бо це суперечить характеристиці графів блоків (Теорема 2.10). Він також не може бути C_3 , бо тоді відповідні блоки вершинам C_3 мають перетинатись по одній і тій же вершині, а отже будемо мати червону вершину, що лежить у 3 блоках (суперечить умові).

\Leftarrow Якщо $B(G)$ – ланцюг, то кожне вершина графа $B(G)$ має степінь ≤ 2 , а тому кожна червона вершина має міститись у рівно 2 блоках (інакше б утворився повний підграф). Аналогічно кожен блок містить щонайбільше 2 червоні, інакше б у нашому ланцюгу була б вершина степеня більше як 2. □

Твердження 2.15. $B(G)$ є деревом тоді й тільки тоді, коли кожна червона вершина G – лежить рівно в 2 блоках.

Доведення. \Leftarrow Оскільки кожна червона лежить у 2 блоках, тоді граф $B(G)$ не міститиме циклів. Адже, виходячи з означення блокового оператора, цикли можуть утворитись лише разом із утворенням K_n , $n \geq 3$, на місцях перетину більше як 2 блоків.

\Rightarrow Від супротивного, припустимо, існує червона вершина у графі G степеня більше як 2. Тоді відповідні блокам вершини, яким належить ця вершина утворять повний граф K_n , $n > 2$ у графі $B(G)$ – маємо суперечність. □

Твердження 2.16. *Якщо T - дерево, то $B^{2n}(T)$ - теж дерево.*

Доведення. Нехай маємо T – дерево. Якщо T – є ланцюгом, то це тривіальний випадок. Тому розглядатимемо випадок, що граф T має принаймні одну вершину степеня $n \geq 3$. Тоді на цьому місці у графі $B^1(T)$ матимемо повний підграф K_n . Оскільки це дерево, то кожен його блок – це ребро. А тому кожна відповідна вершина v ребру uw лежатиме рівно у 2 блоках: один блок утворений із вершин, що відповідають ребрам інцидентним вершині u , а інший відповідно вершині w . Тому кожна червона вершина підграфа K_n лежить у рівно 2 блоках. Тоді використаємо Твердження 2.15, що завершує доведення. \square

Означення 2.17. $\beta_1(G) = \min\{n \in \mathbb{Z}_+ : B^n(G) \text{ є ланцюгом}\}$.

Означення 2.18. Покладемо $\beta_2(G) = \min\{n \in \mathbb{Z}_+ : B^n(G) \text{ є деревом}\}$;

Твердження 2.19. *Якщо T - дерево, то застосувавши два рази блоковий оператор видаляться висячі вершини. $B^2(T) \simeq T$ -висячі вершини.*

Доведення. Нехай маємо T – дерево. Якщо T – є ланцюгом, то його довжина скоротиться на 2 (за Теоремою 2.12). Таким чином кожний ланцюг у дереві скоротиться у 2 рази (тобто видяться кінці, висячі вершини). А за Твердженням 2.16 $B^2(T)$ є деревом, що завершує доведення. \square

Алгоритм знаходження $\beta_1(T)$, де T – дерево.

У дереві для кожних двох діаметральних ланцюгів $P_{1,2}$ маємо $d(P_1) = d(P_2)$, де $d(P) = \max\{d(u, v) : u \in P, v \notin P\}$. Визначаємо число $d(G)$ для графа блоків як $d(P)$ для деякого діаметрального ланцюга P . Легко показати, що $\beta_1(T) = 2d(T)$. За Твердженням 2.19 кожні дві ітерації блокового оператора обрізатись висячі вершини. Таким чином граф $B^{2d(T)}(T)$ всі ланцюги меншої довжини за діаметральний знищуються. А отже, отримаємо ланцюг.

Наслідок 2.20. $\beta_1(G) = \beta_2(G) + \beta_1(B^{\beta_2(G)}(G))$

Доведення. Напрямую впливає і означень β_1 та β_2 . \square

Питання про оцінки β_2 для довільного графа та β_2 взагалі залишається відкритими. Ітерації блокового оператора можна розглядати як динамічну систему. На місцях вершин, по яких перетинаються більше як 2 блоки виникають повні підграфи, наступною ітерацією вони зникають. Хоча для графа блоків також виконується умова про перетин двох діаметральних ланцюгів, питання також потребує подальших досліджень. Можна дати лише верхню оцінку, як $\beta_1(G) \leq \beta_1(T_G)$, де T_G – кістякове дерево нашого графа блоків G .

2.2.4 Опис алгоритмів із доданого коду

Алгоритм побудови реберного графа:

Алгоритм запрограмовано мовою програмування *Haskell*

```
type Graph = [[Int]]
```

Граф задається списком списків цілих чисел. На i -тій позиції списку міститься список суміжних вершин до вершини v_i .

Функція **toGraphE** переводить граф в представлення через ребра (список ребер). Тобто це список, в якому зберігаються списки довжини 2 - це пари вершин, тобто ребро.

Функція **addNewEdges** додає нові ребра до графа, в собі викликає функцію `addEdge`, що додає одне ребро.

Функція **edgeGraphE** буде сам граф перетинів, на вхід приймає граф у реберному вигляді. Має складність $O(N^2)$, оскільки виконує прохід по всіх ребрах, щоб їх додати, а функція додавання ребер в собі виконує пошук суміжних вершин до обох кінців ребра.

Алгоритм знаходження точок зв'язності[10]: Алгоритм проілюстровано мовою програмування *C++*, має складність $O(V + E)$.

Алгоритм побудований на основі проходження по графу в глибину (DFS). При проходженні (DFS) у нас утворюється дерево обходу (DFS tree). Воно має таку структуру: вершина u є батьківською для дочірньої вершини v , якщо вершина v була відвідана із вершини u .

Тоді в представленні графа за допомогою дерева DFS, верша u є точкою зв'язності, якщо виконується одна з двох умов:

1) u - корінь дерева, що має принаймні 2 дочірні вершини.

2) u не корінь дерева, що має дочірню вершину v таку, що жодна вершина в піддереві з коренем v не є суміжною з жодною із батьківських вершин v .

При виконанні обходу дерева у масив `parent[]` за індексами вершин записуються їх батьківські вершини. Щоб перевірити перший випадок, потрібно, щоб вершина u не мала батьківської (`parent[u] == -1`) та, щоб u мала принаймні дві дочірні вершини. Щоб перевірити другий випадок ми тримаємо масив `disc[]`, що записує час приходу до вершини. Для кожної вершини зберігається мінімальний час приходу до неї. Також для кожної вершини u ми маємо віднайти найбільш ранню пройдену вершину в піддереві з коренем u . Тому ми тримаємо додатковий масив `low[]`. Він оновлюється таким чином: $low[u] = \min(disc[u], disc[w])$, де w - предок u та існує вершина між нащадком u та w .

Таким чином функція **AP()** виконує обхід дерева, викликаючи в собі функцію **APUtil()**, що виконує оновлення вище описаних масивів та перевірок 2 умов.

Алгоритм знаходження блоків[7]:

Алгоритм в собі використовує вище описаний алгоритм знаходження точок

зв'язності. Він виконує обхід графа в глибину. Починаємо з довільної вершини, і збираємо компоненту зв'язності в список. Якщо вершина не є точкою зв'язності, то приєднуємо всіх її сусідів, а якщо є, то не приєднуємо.

Функція **ВСС()** виконує обхід графа, а функція **ВССUtil()** перевірку умов та оновлення масивів. Має складність $O(V + E)$.

Алгоритм побудови графа блоків: Спочатку знаходимо компоненти двозв'язності – наші блоки. А потім будуємо граф блоків. Виконуємо перевірку, чи перетинаються множини вершин блоків. Тут є декілька підходів. Наївний підхід - перевірка усіх пар множин - це $O(N^2)$, та виконання перетину множин - залежно від використаної структури даних, в кращому випадку - $O(N)$ (використовуючи хешування). Тому маємо $O(E + V + N * B^2)$, де B - кількість блоків, N - розмір блока. Цей підхід можна покращити. При знаходженні блоків потрібно запам'ятати точки зв'язності. І тоді замінити перевірку перетин множин, перевіркою, чи містить пара блоків спільну точку зв'язності. Тоді матимемо $O(E + V + A * B^2)$, де A - кількість компонент зв'язності. Враховуючи, що випадкове дерево має $\frac{V}{e}$ - листочків, де e число Ойлера та, що кожна не висяча вершина є точкою зв'язності. У випадку, якщо наш граф є деревом матимемо: $O(E + V + (V - \frac{V}{e}) * E^2)$.

Знаходження $\beta_0, \beta_1, \beta_2$: Для знаходження кожного β_i ітеративно будуємо блокові графи, виконуючи відповідну перевірку. Для β_0 - чи не є наш граф K_1 , тобто має одну вершину. β_1 - це перевірка на дерево, виконуємо обхід графа, і перевіряємо, чи не потрапляємо в цикл, не проходимо другий раз по вже відвіданій вершині. β_2 - перевірка на ланцюг, аналогічно як із деревом, але маємо додаткову перевірку, що кожна вершина має ≤ 2 суміжні.

2.3 Реберні графи

2.3.1 Означення

Означення 2.21. Нехай G граф. Його *реберним графом* $L(G)$ називається граф перетинів $\Omega(V(G), E(G))$. Таким чином, вершинами графа $L(G)$ є ребра G , і два ребра суміжні в $L(G)$, якщо вони мають спільну вершину.

Приклад 2.22. Приклад реберного графа з рисунком.

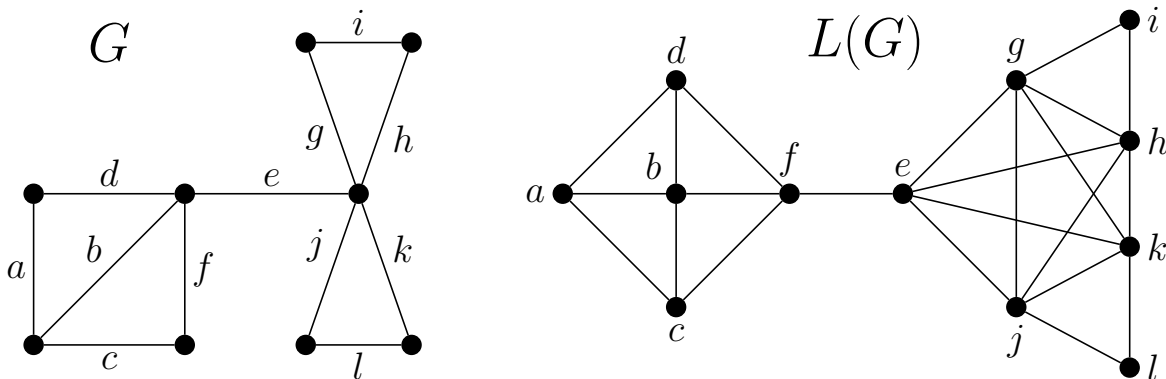


Рис. 2.9. Граф G , та його реберний граф $L(G)$

Твердження 2.23. Для кожного графа G має місце рівність:

$$|E(L(G))| = \sum_{v \in V(G)} C_{deg(v)}^2$$

Доведення. Нехай маємо граф G , позначимо $V(G)$ та $E(G)$ – множини його вершин і ребер відповідно. Для його реберного графа аналогічно: $L(G): V(L(G)), E(L(G))$. Зауважимо, що $|V(L(G))| = |E(G)|$ – за означенням реберного графа. Зафіксуємо вершину $p \in V(G)$. Позначимо $I_p = \{e_1, e_2, \dots, e_n\}$ – множини інцидентних їй ребер у графі G , $deg(p) = |I_p|$. Позначимо $J_p = \{v_1, v_2, \dots, v_n\}$ – множини відповідних їм вершин у графі $L(G)$. Зауважимо, що будь-яка пара вершин $v_i, v_j \in J_p$ з'єднана ребром, оскільки мають спільну вершину p . Таким чином реберний граф міститиме $C_{|J_p|}^2 = C_{|I_p|}^2 = C_{deg(p)}^2$ комбінацій пар ребер, що інцидентні вершині p графа G . Отже, щоб знайти $|E_{L(G)}|$ потрібно знайти суму таких комбінацій всіх вершинах графа G :

$$|E(L(G))| = \sum_{v \in V(G)} C_{deg(v)}^2$$

□

Теорема 2.24. Для графа G має місце ізоморфізм $G \simeq L(G)$ тоді й тільки тоді, коли G простий цикл.

Доведення. \Leftarrow Нехай задано граф-цикл $G = C_n$. Позначимо його множини вершин і ребер відповідно: $V = \{v_1, v_2, \dots, v_{n-1}, v_n\}$ та $E = \{e_1, e_2, \dots, e_{n-1}, e_n\}$, де e_i з'єднує вершини v_i та $v_{(i+1) \bmod n}$. Тоді $V(L(G)) = \{e'_1, e'_2, \dots, e'_{n-1}, e'_n\}$, де e'_i відповідає реброві e_i графа G . $E(L(G)) = \{q_1, q_2, \dots, q_{n-1}, q_n\}$.

Тоді кожна вершина e'_i графа $L(G)$ матиме рівно 2 суміжні вершини: $e'_{i-1 \bmod n}$ та $e'_{i+1 \bmod n}$. Адже відповідні їм ребра у графі G мали по 1 спільній вершині: $e_{i-1 \bmod n}$ та e_i мали спільну вершину $q_{i-1 \bmod n}$, e_i та $e_{i+1 \bmod n}$ мали спільну вершину q_i . Таким чином маємо ізоморфізм $G \simeq L(G)$, оскільки $\forall i : v_i \simeq e'_i, v_i \simeq q_i$.

\Rightarrow Від супротивного, нехай існує граф G - не цикл такий, що $G \simeq L(G)$. Тоді в графі G існує вершина v така, що $\deg(v) \neq 2$. Тоді $\deg(v) = 1$ або $\deg(v) \geq 3$.

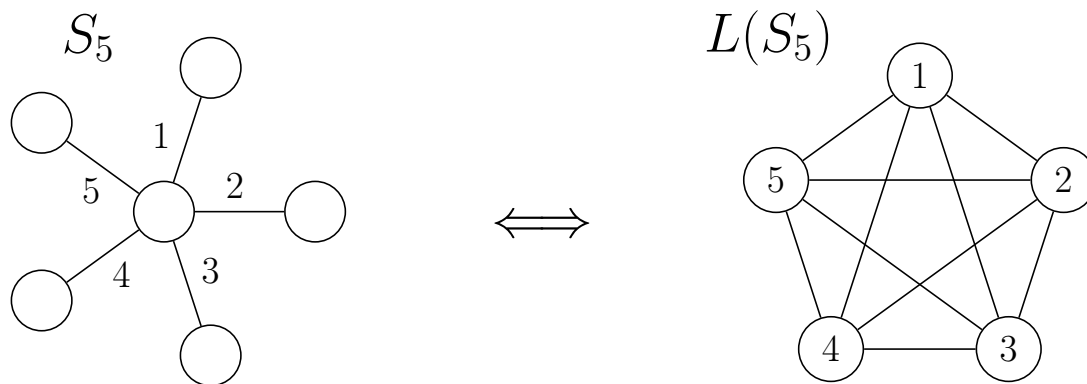
Якщо граф матиме лише вершини степеня 1 та 2, і не є циклом то він є ланцюгом. Щоб це показати почнемо з висячої вершини і будемо йти по ланцюгу до її суміжної. Таким чином кожна наступна вершина буде новою. А якби ми в якийсь момент потрапили у вже відвідану вершину, то її степінь був би 3. Оскільки граф скінчений, то на кінці має бути вершина степеня 1. Тому G – ланцюг. Цей випадок нам не підходить, оскільки він не є ізоморфним своєму реберному графові, адже $L(P_n) = P_{n-1}$. Отже, граф G містить принаймні одну вершину $s_1 : \deg(s_1) \geq 3$.

Оскільки графи G та $L(G)$ є ізоморфними, то вони мають однакову кількість ребер та вершин $|V(G)| = |V(L(G))|$, $|E(G)| = |E(L(G))|$. Також $|E(G)| = |V(L(G))|$ – за означенням реберності. Отже, граф G має однакову кількість ребер та вершин. Оскільки він є зв'язним, мінімально потрібно $n - 1$ ребро, щоб зв'язати всі вершини (створити дерево), крім цього залишається 1 ребро. Отже, граф G має вид $T + e$. Тобто дерево з додаванням одного ребра. Таким чином граф G буде мати рівно один цикл.

Але, якщо G не є циклом, то граф $L(G)$ матиме принаймні 2 цикли: перший – породжений структурою $T + e$, другий з'явиться з вершини $s_1 : \deg(s_1) \geq 3$. Інцидентні їй ребра утворять K_n . Оскільки $n \geq 3$, то у реберному графі виникне ще один цикл.

Таким чином граф G має рівно 1 цикл, а граф $L(G)$ принаймні 2. Маємо суперечність ізоморфності графів. \square

Лема 2.25. *Для графа G з $n \geq 4$ вершинами $L(G) \simeq K_n$ тоді і тільки тоді, коли $G \simeq S_n$.*

Рис. 2.10. $K_5 \simeq L(S_5)$

Доведення. \Leftarrow Кожні два ребра зірки мають спільну центральну вершину, тому кожна відповідна їм пара вершин у реберному графові буде суміжною. Отже, у $L(G)$ будь-які дві вершини попарно з'єднані, тому і матимемо повний граф. Оскільки зірка S_n має n ребер, то $L(S_n)$ матиме n вершин, отже $L(S_n) \simeq K_n$.

\Rightarrow Нехай маємо граф $K_n \simeq L(G)$, $n \geq 4$. Зафіксуємо вершину $v \in V(L(G))$, позначимо u_1u_2 – відповідне їй ребро у графі G . Тоді оскільки $L(G)$ – повний граф. То ребро u_1u_2 є суміжним до всіх інших ребер. Якщо всі інші ребра мають з ним одну спільну вершину, нехай u_1 , то тоді маємо зірку $G \simeq S_n$, де вершина u_1 є її центром.

Тоді залишається лише випадок, коли одна частина ребер суміжні по u_1 , а інша по u_2 . За принципом Діріхле, на одну з вершин u_1, u_2 припадуть два мінімум два ребра. Без втрати загальності нехай це u_1 , тоді $\deg(u_1) \geq 3$, $\deg(u_2) \geq 2$ (з врахуванням ребра u_1u_2). Оскільки $L(G)$ – повний граф, то $\forall x, y : x \in A(u_1), y \in A(u_2)$ x, y – суміжні, де $A(v)$ позначає множину суміжних вершин до вершини v . Оскільки $|E(G)| = |V(L(G))|$, то нових ребер додавати ми не можемо. Таким чином $A(u_1) = A(u_2) = z$ а отже матимемо мультиграф, бо $n - 2$ суміжні вершини до вершин u_1, u_2 є однією і тією вершиною. □

Теорема 2.26. Уїтні (1932) Нехай G_1, G_2 два зв'язні графи такі, що їхні реберні графи $L(G_1) \simeq L(G_2)$ ізоморфні. Тоді $G_1 \simeq G_2$ за виключенням пари $\{G_1, G_2\} = \{K_3, K_{1,3}\}/3$.

Лема 2.27. Нехай маємо реберний граф $L(G)$, тоді для кожної вершини $v \in V(L(G))$ виконується наступне. Нехай A_v – множина її суміжних ребер. Будемо розбивати множину суміжних їй ребер на множини T_1, T_2, \dots, T_n таким чином, щоб будь-яка пара вершин взятих з різних множин не була суміжною. $\forall r \in T_i, q \in T_j, i \neq j : r$ та q не суміжні. Тоді максимальна кількість таких множин T_k дорівнює 2.

(інакше кажучи) Максимальна кількість множин на яку можна розбити суміжні вершини, кожної вершини графа $L(G)$, так щоб ніякі дві вершини у різних множинах не були суміжні це 2.

Доведення. Від супротивного, припустимо, що у графові $L(G)$ є вершина x , для якої таке число більше ніж 2. Не втрачаючи загальності, покажемо, що розбиття на 3 такі множини не може існувати, а отже і на більшу кількість. Припустимо, ми розбили суміжні вершини x на 3 такі множини:

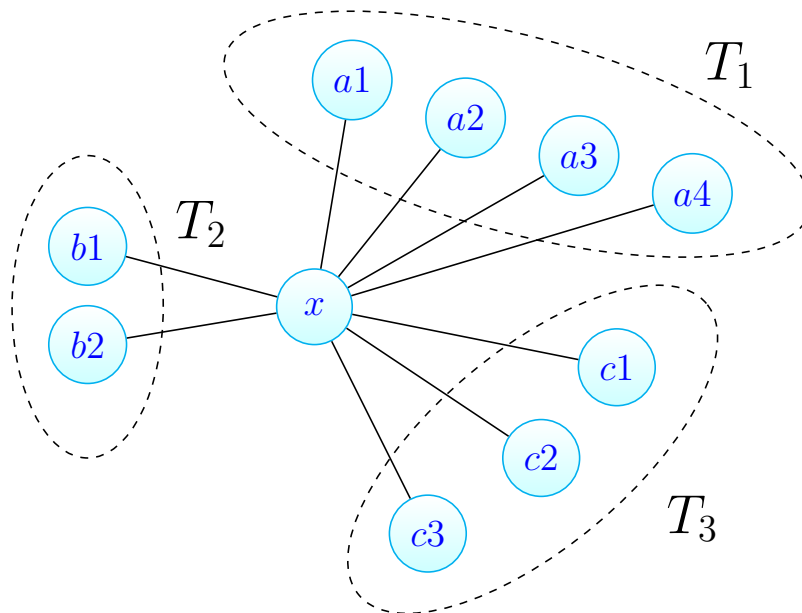


Рис. 2.11. Ілюстрація 1 до Лема 2.27

Маємо такі множини: $T_1 = \{a_n\}$, $T_2 = \{b_n\}$, $T_3 = \{c_n\}$. Зафіксуємо по 1 елементу з кожної множини, нехай a_1, b_1, c_1 . Зафіксуємо ребра у графі G , які відповідають цим вершинам: $e_{a_1}, e_{b_1}, e_{c_1}$. Також позначимо ребро e_x , що відповідає центральній вершині. Ребра x у графі G має дві вершини, позначимо як v_1 та v_2 . Оскільки кожна із зафіксованих вершин у графі $L(G)$ суміжна із x , то відповідні їм ребра графа G мають спільні вершини. Для кожного із відповідних ребер $e_{a_1}, e_{b_1}, e_{c_1}$, є 2 варіанти, як прикріпитися до ребра x : v_1 та v_2 . Маємо класичний принцип Діріхле. Тому знайдеться така вершини v_i , на яку прикріпиться мінімум 2 ребра. Тоді при переході до реберного графа ці дві вершини будуть суміжними в графі $L(G)$, адже мають спільну вершину v_i , що суперечить вибору множин T_n .

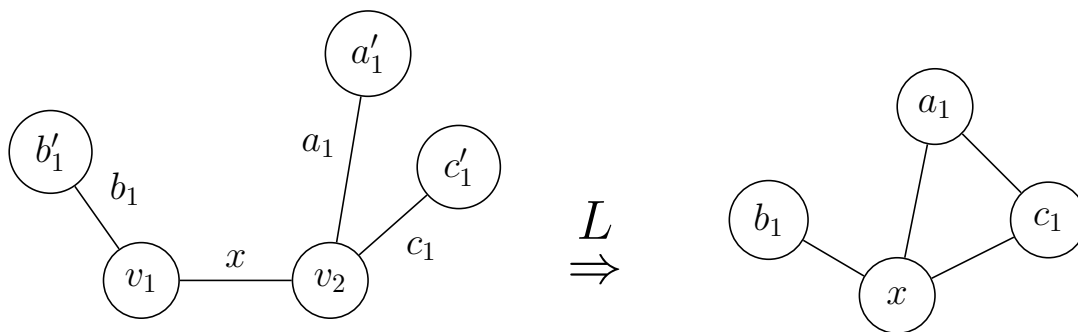


Рис. 2.12. Ілюстрація 2 до Лема 2.27

Якщо не можна розмістити навіть 3 ребра на двох вершинах, то тим паче не можна і більше. Тому максимум такого розбиття – це дві множини. □

Наслідок 2.28. Кожна з множин T_n разом з вершиною x у графі $L(G)$ утворюють повний підграф $K_{|T_n|+1}$.

Доведення. Оберемо вершину x графа $L(G)$ та позначимо v_1, v_2 інцидентні вершини відповідного їй ребра у графі G . Тоді за лемою розіб'ємо множину її суміжних вершин на дві множини.

Позначимо $T(v_1)$ – ребра, що інцидентні вершині v_1 , ребра x у графові G . Та $T(v_2)$ для вершини v_2 відповідно. Зауважимо, що x не є елементом жодної з множин.

Якщо кількість таких множин 0, то маємо односточковий граф K_1 .

Якщо кількість таких множин 1, то тоді всі вершини є суміжними до однієї з $\{v_1, v_2\}$. Без втрати загальності, нехай це v_1 . Тоді всі ребра з множини $T(v_1)$ мають спільну вершину v_1 , а тому відповідні їм вершини у графові $L(G)$ будуть попарно суміжними. Отже, матимемо повний граф $K_{|T(v_1)|+1}$ (+1 для самого ребра x).

А якщо обидві множини T_1, T_2 не порожні, то тоді всі ребра однієї множини, нехай, T_1 є суміжними до v_1 , а всі ребра іншої T_2 суміжними до v_2 . Через те, що всі кожної множини (окремо) мають по спільній вершині v_1 та v_2 відповідно. Граф $L(G)$ матимемо два повних підграфи $K_{|T(v_1)|+1}$, та $K_{|T(v_2)|+1}$ зі спільною вершиною x . □

2.3.2 Характеризація

Теорема 2.29. Для графа H існує граф G із $L(G) \simeq H$ тоді й тільки тоді, коли множину ребер H можна розбити на повні підграфи так, щоб кожна його вершина належала не більше ніж двом цим підграфам.

Доведення. \Rightarrow Застосуємо наслідок 3.12 для кожної вершини v_i графа $L(G)$ таким чином, множину суміжних вершин для кожної v_i можна розбити на 1-2 множини, які разом із самою вершиною v_i та інцидентними їй ребрами утворюють повні

графі. З леми на кожній такій вершині утвориться максимум 2 таких повних графа, яким вона буде належати, що і треба було показати.

⇐ Покажемо, що наступний алгоритм будує граф $G : L(G) \simeq H$.

1) Виконаємо розбиття, щоб кожна його вершина належала не більше ніж двом цим підграфам. Нехай маємо родин підмножин

$$\mathcal{F} = \{\{a_1, \dots, a_n\}, \{b_1, \dots, b_m\}, \dots\}$$

. Кожен елемент \mathcal{F} є множиною вершин повного підграфа графа G .

2) Поставимо у відповідність кожному повному підграфу K_n зірку S_n . Отримаємо множину

$$\mathcal{S} = \{\{a'_0, a'_1, \dots, a'_n\}, \{b'_0, b'_1, \dots, b'_n\}, \dots\}$$

. Вершини з індексами 0 позначають центральну вершину зірки S_n , а інші відповідні вершини графа G ($a_k \sim a'_k, b_k \sim b'_k, \dots$).

3) Якщо вершина у графі G належить одному повному підграфу, то ми її не чіпаємо, а якщо належать двом, то видаляємо відповідні їм вершини із зірок і додаємо ребро між центрами відповідних зірок.

Наприклад, у граф G вершина w_k належить 2 повним графам Q, P при розбитті. Позначимо відповідні їм зірки, отримані у кроці 1 як $S(Q)$ та $S(P)$. Також вершини q_k, p_k , що відповідають вершині w_k . Тоді видаляємо вершини q_k, p_k (разом з ребрами до центрів $q_k q_0, p_k p_0$) і додаємо ребро $q_0 p_0$, що сполучає центри графів $S(Q)$ та $S(P)$. Тоді

Наведемо приклад роботи алгоритму: нехай маємо такий граф $L(G)$:

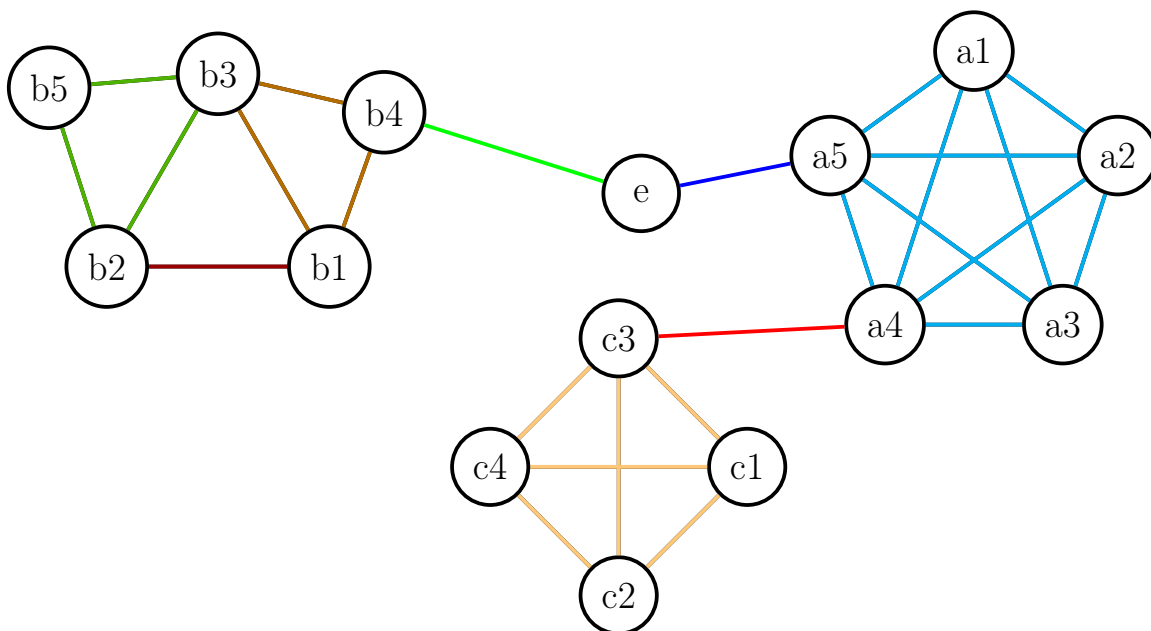


Рис. 2.13. Приклад, розбиття на повні підграфи

Далі на місці кожного повного підграфа ставимо зірку. По центру ставимо вершину (центральною) і від неї проводимо промені до місць, де були вершини.

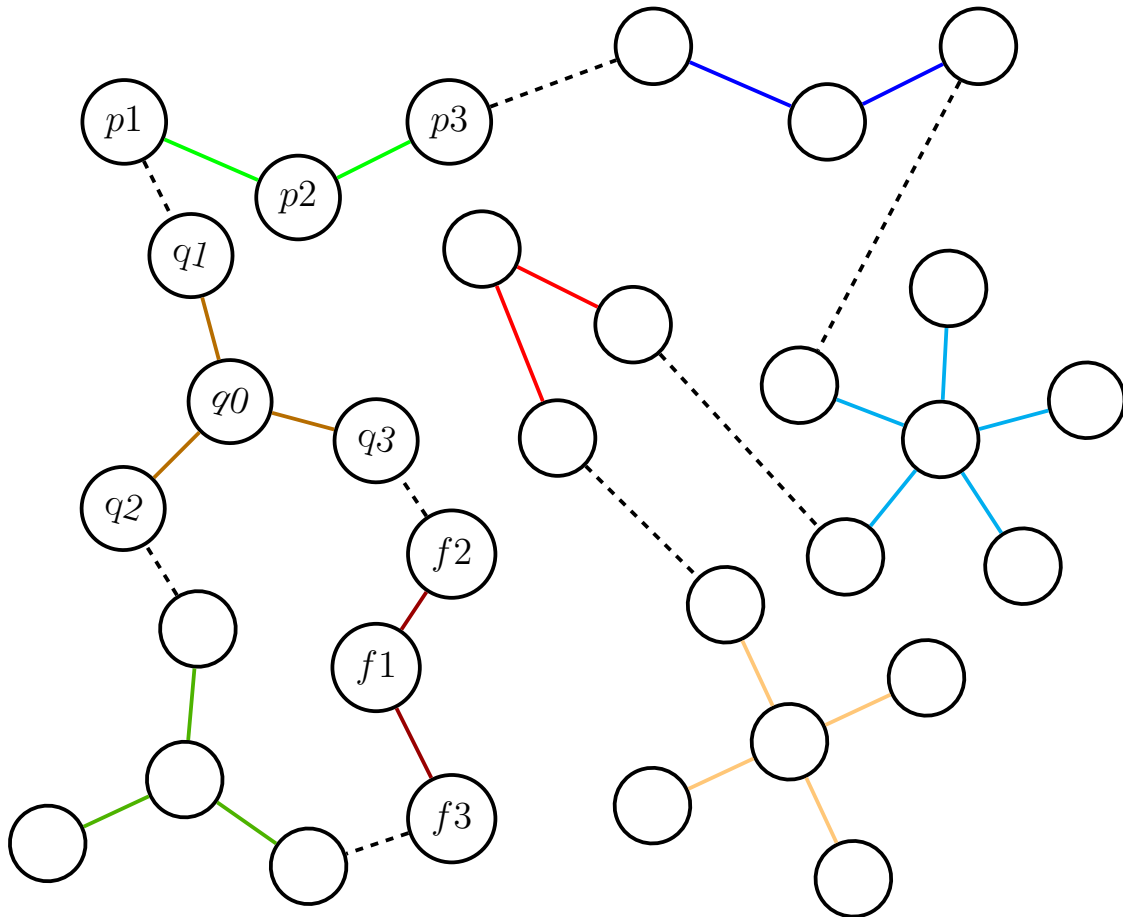
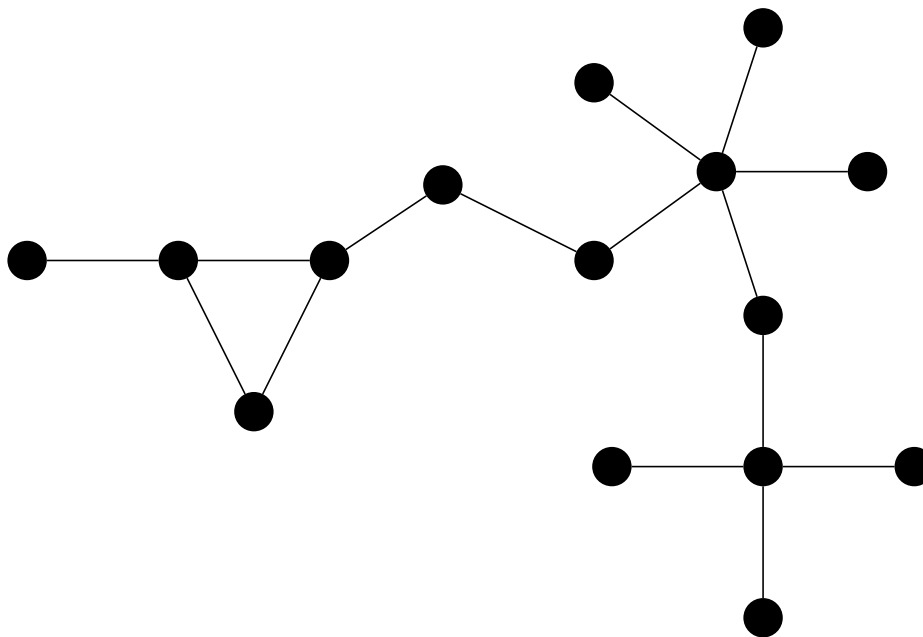


Рис. 2.14. Наступний крок – заміна на зірки

Пунктиром позначено зв'язки між однією і тією ж самою вершиною в різних зірках.

Фінальним кроком є вилучення пунктирних зв'язків. Робити це потрібно наступним чином, видаляємо вершини, з'єднані пунктиром і напямую з'єднуємо їх суміжні вершини. Наприклад у ланцюгу $\{q_0 - q_1 - p_1 - p_2\}$ вершини q_1 та p_1 з'єднання пунктиром, то ми їх видаляємо і безпосередньо сполучаємо q_0 з p_2 . Ще приклад: $\{q_0 - q_3 - f_3 - f_1\}$ перейде у $\{q_0 - f_1\}$.

У результаті граф G матиме вигляд:

Рис. 2.15. Граф G

□

Лема 2.30. $B(G) \simeq L(G)$ тоді й тільки тоді, коли G – дерево.

Доведення. \Leftarrow Оскільки T – це дерево, то кожне його ребро є окремим блоком. Таким чином, з кожного ребра в обох випадках отримуємо по одній вершині: $V(B(T)) = V(L(T))$. Множини ребер теж є однаковими за означеннями графа блоків та реберного графа (з точки зору реберного: вершини, що відповідають суміжним ребрам будуть сполучатись, а з точки зору графа блоків, ці суміжні ребра і будуть блоками, зі спільною точкою дотику).

\Rightarrow Нехай маємо $B(G) \simeq L(G) = H$, тоді з ізоморфізму маємо $V(B(G)) \simeq V(L(G)) \simeq E(G)$. Тоді з означень: кожне ребро графа G є окремим блоком і відповідно кожен блок є окремим ребром. А отже, якщо кожне ребро графа G є окремим блоком, то граф G не містить циклів (якби він містив цикл, то цей цикл був би окремим блоком, що суперечило б тому, що кожне ребро є окремим блоком). □

Твердження 2.31. Для графа H існує дерево T із $L(T) \simeq H$ тоді й тільки тоді, коли H є зв'язним графом блоків, кожна вершина якого належить не більше ніж двом його блокам.

Доведення. \Rightarrow При побудові реберного графа $L(T)$. На кожній вершині v дерева T утвориться повний граф K_n , де $n = \text{deg}(v)$. Тому це і буде графом блоків. За лемою 3.13 будь-яка його вершина належить максимум 2 таким блокам.

\Leftarrow Покажемо, що за даних умов алгоритм з теореми 3.6 буде дерево. Оскільки кожна вершина належить не більше ніж двом його блокам, то $B(H)$ не містить

циклів (є деревом). Додавши відповідні кількості висячих вершин, ми отримуємо граф $T : B(T) \simeq H$. Граф T є деревом, бо додавання висячих не призводить до появи циклів. Тоді за Лемою 3.15 $L(T) \simeq B(T) \simeq H$. \square

Твердження 2.32. Для дерева T існує граф G такий, що $L(G) = T$ тоді і тільки тоді, коли G є ланцюгом.

Доведення. \Rightarrow Оскільки $L(G) = T$, то граф G не містить циклів, і також не містить вершин степеня більш як 2. Інакше $L(G)$ містив би цикли. Таким чином G – ланцюг.

\Leftarrow Якщо $G = P_n$ – (ланцюг на n вершинах), то $L(G) = P_{n-1}$. \square

Означення 2.33. Трикутником з вусиками будемо називати граф, що отримується із K_3 приклеюванням рівно одного ланцюга до кожної з трьох вершин.

Приклад 2.34. Приклад трикутника з вусиками

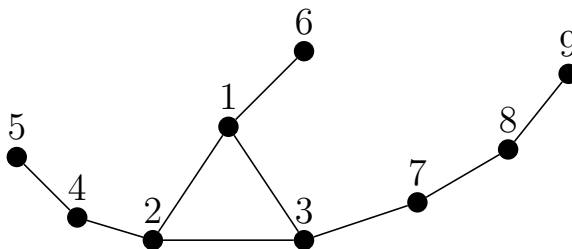


Рис. 2.16. Трикутник з вусиками

Означення 2.35. Великою клешнею будемо називати дерево з єдиною вершиною степеня 3, а інші вершини мають степені 1,2.

Приклад 2.36. Приклад великої клешні

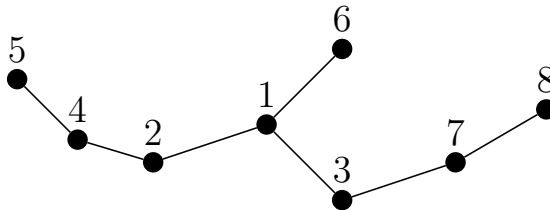


Рис. 2.17. Велика клешня

Твердження 2.37. Нехай граф G – зв'язний та уніциклічний, тоді існує граф H такий, що $L(H) = G$, тоді й тільки тоді, коли G – це C_n або трикутник з вусиками.

Доведення. \Rightarrow Якщо граф G – уніциклічний, то граф H не може мати більш як 1 циклу. Якщо граф H містить рівно один цикл, то він має бути тільки простим циклом C_n (якщо додати ще одну вершину, то з'явиться вершина степеня 3, таким чином $L(H)$ матиме 2 цикли). Другий випадок, коли H не має циклів, тоді він не може мати вершин степеня більше за 3 (інакше реберний матиме кілька циклів, адже утвориться підграф K_n , $n \geq 4$). Такий граф матиме рівно одну вершину степеня 3, а інші вершини степенів 1,2 – тобто є великою клешнею.

$\Leftarrow L(C_n) = C_n$ (уніциклічний). $L(\text{велика клешня}) = \text{трикутник з вусиками}$ (теж уніциклічний). \square

2.4 Графи клік

2.4.1 Означення

Означення 2.38. *Клікою* називається підмножина множини вершин графа така, що кожні дві вершини з цієї підмножини з'єднані ребром та яка не може бути розширена додаванням однієї із суміжних вершин. Це еквівалентно тому, що підграф, породжений цією підмножиною є повним та не міститься в іншому більшому повному підграфі.

Означення 2.39. *Клікове покриття графа* – це розбиття множини його вершин на підмножини, так що:

- 1) кожна підмножина є повним підграфом;
- 2) кожне ребро графа належить принаймні одній з цих підмножин.

Приклад 2.40. Приклад, графа та його клікового покриття:

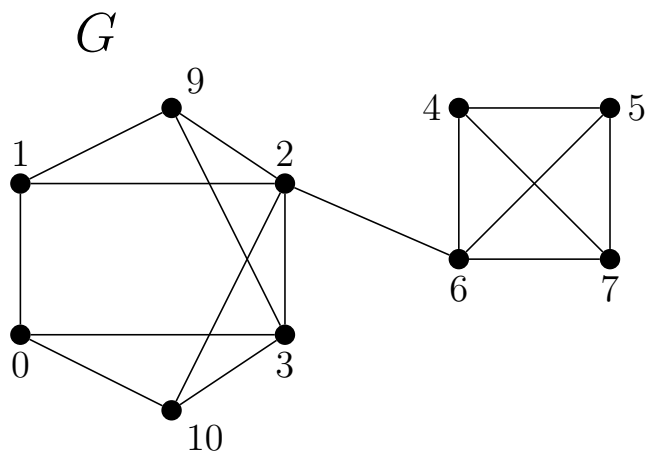


Рис. 2.18. Граф G , що має клікове покриття

$$\mathcal{E} = \{\{1, 9, 2\}, \{0, 3, 10\}, \{9, 3, 2\}, \{10, 2, 3\}, \{0, 1\}, \{2, 6\}, \{4, 5, 6, 7\}\}$$

Твердження 2.41. *Нехай Q та Q' – дві кліки графа G і v – вершина, $v \in Q$ та $v \notin Q'$, тоді існує вершина v' така, що $v \notin Q$ та $vv' \notin E(G)$.*

Доведення. Нехай маємо граф G , дві кліки Q, Q' та вершину v як в умові. Одразу зауважимо, що вершина v' не може належати $Q \cap Q'$, інакше б $vv' \in E(G)$, як дві вершини з кліки Q . Від супротивного, припустимо, що такої вершини не існує, тоді кожна пара вершин $v \in Q, v' \in Q'$ з'єднана ребром. Але тоді $Q \cup Q'$ – клікою, що суперечить максимальності клік Q та Q' . □

Означення 2.42. Цикл називається *породженням*, якщо він є породженим підграфом.

Твердження 2.43. *Граф клік $K(G)$ є двочастковим тоді й тільки тоді, коли жодна вершина графа G не належить більше ніж 2 клікам та G не містить непарних породжених циклів довжини більше як 3.*

Доведення. \Rightarrow Від супротивного, припустимо існує такий граф G , що $K(G)$ – двочастковий, і умова не виконується. По-перше, зауважимо, що якщо в графі G існує вершина, що належить $n \geq 3$ клікам, то граф $K(G)$ міститиме повний підграф K_n , а тому і не може бути двочастковим. Тепер припустимо, що G містить непарний породжений цикл C_n , $V(C_n) = \{v_1, \dots, v_n\}$, $E(C_n) = \{e_1, \dots, e_n\}$, $n \geq 5$. Оскільки цикл породжений, то жодні два його ребра не можуть лежати в єдиній кліці. Тому, кожне ребро e_i належить окремій кліці Q_i , а кожна вершина належить клікам двох ребер інцидентних їй. Тому граф $K(G)$ буде містити такий же цикл довжини n . Оскільки n непарне, то маємо суперечність тому, що двочастковий граф не може містити непарних циклів.

\Leftarrow Нехай маємо граф, G як в умові, покажемо, що $K(G)$ є двочастковим. Оскільки кожна вершина належить не більше ніж 2 клікам, то в результаті застосування оператора клік, на місці вершин не утворюватиметься повних графів K_n , $n \geq 3$. Таким чином не утворюватиметься циклів. А оскільки граф G не містить непарних породжених циклів, то граф $K(G)$ їх теж не міститиме, адже кожен парний породжений цикл графа G перейде в парний породжений цикл графа $K(G)$ (як було показано в доведенні необхідності). \square

Означення 2.44. Нехай G – граф із множиною вершин $V(G) = \{v_1, \dots, v_n\}$ та кліковим покриттям $\mathcal{E} = \{Q_1, \dots, Q_k\}$. Дуальним представленням \mathcal{F} множини \mathcal{E} будемо називати родину $\mathcal{F}(\mathcal{E}) = \{S_1, \dots, S_n\}$, де $S_i = \{j : v_i \in Q_j\}$, для кожного $i \in \{1, \dots, n\}$. Таким чином кожна S_i є множиною цілих чисел та $S_i \cap S_j \neq \emptyset$ тоді і тільки тоді, коли $v_i, v_j \in E(G)$.

Наслідок 2.45. *Нехай G – граф, $\mathcal{E} = \{Q_1, \dots, Q_n\}$ – його клікове покриття, $\mathcal{F} = \mathcal{F}(\mathcal{E})$ – дуальне представлення. Тоді $G \simeq \Omega(\mathcal{F})$.*

Доведення. Результат напряду випливає з означення \mathcal{F} , адже кожна множина $F_i \in \mathcal{F}$ міститиме множину індексів клік, яким належить вершина i графа G . Тому, якщо вершини суміжні, то вони містять в якійсь кліці і будуть з'єднані ребром у графі перетинів, а не суміжні вершини відповідно не містяться разом в жодній кліці й тому не будуть з'єднані ребром в графі перетинів. \square

Твердження 2.46. *Нехай G – граф та $\mathcal{E} = \{Q_1, \dots, Q_m\}$ – його клікове покриття. Тоді дуальне представлення $\mathcal{F} = \mathcal{F}(\mathcal{E})$ є кліковим покриттям графа перетинів $H \simeq \Omega(\mathcal{E})$.*

Доведення. Нехай маємо G , \mathcal{E} , \mathcal{F} , H – як в умові.

1) Зафіксуємо довільну множину $S_k \in \mathcal{F}(\mathcal{E})$, тоді $\forall i, j \in S_k : ij \in E(H)$ тому, що відповідні кліки Q_i та Q_j перетинаються, оскільки принаймні $v_k \in Q_i \cap Q_j$. Отже, кожна множина родини \mathcal{F} є повним підграфом графа H .

2) Візьмемо довільне ребро $ij \in E(H)$, тоді відповідні йому кліки перетинаються $Q_i \cap Q_j \neq \emptyset$. Тому існує $v_k \in Q_i \cap Q_j$. А отже $i, j \in S_k \in \mathcal{F}(\mathcal{E})$, таким чином множина S_k накриває ребро ij . \square

Означення 2.47. Нехай $\mathcal{F} = \{S_1, \dots, S_n\}$ – родина підмножин деякої множини S . Говорять, що \mathcal{F} задовільняє умові Хеллі, коли виконується наступна умова: якщо всі елементи підродини $\mathcal{F}' \subseteq \mathcal{F}$ попарно перетинаються ($S_i, S_j \in \mathcal{F}' : S_i \cap S_j \neq \emptyset$), то загальний перетин всіх елементів цієї підродини \mathcal{F}' не порожній ($\cap\{S_i : S_i \in \mathcal{F}'\} \neq \emptyset$).

Лема 2.48. Нехай G – граф та $\mathcal{E} = \{Q_1, \dots, Q_m\}$ – його клікове покриття. Нехай $H \simeq \Omega(\mathcal{E})$ – граф перетинів. \mathcal{E} задовільняє умову Хеллі тоді і тільки тоді, коли дуальне представлення $\mathcal{F}(\mathcal{E})$ містить кожну кліку графа H .

Доведення. Нехай маємо $G, \mathcal{E}, \mathcal{F}$ та H як в умові. За Твердженням 2.46 \mathcal{F} – клікове покриття графа H . Тому кожна $S_i \in \mathcal{F}$ є повним підграфом H .

\Rightarrow Нехай \mathcal{E} задовільняє умові Хеллі. Зафіксуємо R – довільну кліку графа H та покажемо, що $R \in \mathcal{F}$. Якщо $j, k \in R$, то $jk \in E(H)$ і також $Q_i \cap Q_j \neq \emptyset$. Таким чином всі елементи підродини $\{Q_j : j \in R\}$ родини \mathcal{E} попарно перетинаються. Тоді за умовою Хеллі загальний перетин не порожній, а отже існує спільна вершина $v_i \in \cap\{Q_j : j \in R\}$. Тоді $\forall j \in R : v_i \in Q_j$. А отже $\forall j \in R : j \in S_i = \{s : v_i \in Q_s\}$ та $R \subseteq S_i$. Отже, $R = S_i \in \mathcal{F}$.

\Leftarrow Нехай маємо клікове покриття $\mathcal{F}(\mathcal{E}) = \{S_1, \dots, S_n\}$, що містить кожну кліку графа H . Зафіксуємо підродину $\mathcal{E}' \subseteq \mathcal{E}$, елементи якої попарно перетинаються. Виберемо довільні дві множини $Q_j, Q_k \in \mathcal{E}'$. Тоді існує $v_i \in Q_j \cap Q_k$ та відповідно $j, k \in S_i = \{s : v_i \in Q_s\}$ – за означенням $\mathcal{F}(\mathcal{E})$, і тому $jk \in E(H)$ (будь-яка пара вершин з'єднана ребром). Тоді множина $\{s : Q_s \in \mathcal{E}'\}$ є повним підграфом графа H , а отже містить в деякій кліці $S_q \in \mathcal{F}$. І тоді за означенням $\mathcal{F}(\mathcal{E})$, v_q міститься в кожній $Q_j \in \mathcal{E}'$. \square

Твердження 2.49. Для графа G існує двочастковий граф H такий, що $L(H) \simeq G$ тоді й тільки тоді, коли граф клік $K(G)$ є двочастковим та будь-які дві кліки графа G перетинаються не більш ніж по 1 вершині.

Доведення. \Leftarrow Оскільки будь-які дві кліки графа G перетинаються не більше ніж по одній вершині, то розбиття на кліки і буде розбиттям із характеристизації і за Теоремою 2.29 граф $H : L(G) \simeq H$ справді існує. Зауважимо, що алгоритм побудови графа H , представлений у Теоремі 2.29 аналогічний тому, що взяти граф клік

$K(G)$ і потім до кожної вершини v приєднати стільки висячих вершин, скільки вершин із відповідної кліки вершині v у G вершини знаходяться в одній кліці. Таким чином, оскільки граф $K(G)$ є двочастковим, то отриманий граф після додавання висячих вершин теж буде двочастковим.

⇒ Якщо існує таким двочастковий граф H , що $L(H) \simeq G$, то за тими ж міркуваннями (за алгоритмічною побудовою) $K(G)$ є двочастковим (інакше отримаємо граф H не двочастковий). За характеристизацією реберних графів у відповідному розбитті дві кліки графа G перетинаються не більш ніж по 1 вершині, інакше було б ребро, що потрапляє одразу в дві кліки. □

2.4.2 Оператор клік, характеристизація

Означення 2.50. Визначимо *оператор клік* $K(\cdot)$, таким чином: нехай H – граф, тоді $K(H)$ – граф перетинів на множині всіх клік графа H .

Означення 2.51. Граф G є *графом клік*, якщо існує граф H , такий що $G \simeq K(H)$.

Приклад 2.52. Приклад графа клік

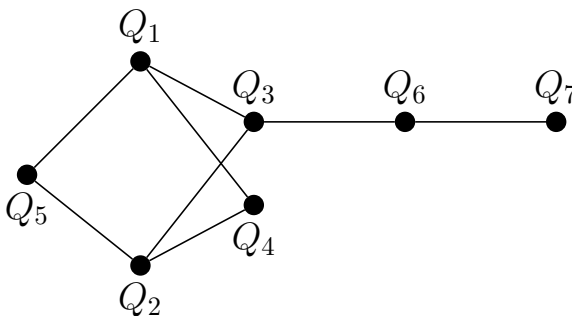


Рис. 2.19. Граф клік $K(G)$ побудований із Прикладу 2.40 (див. рис. 2.18)

Теорема 2.53. *Граф є графом клік тоді й тільки тоді, коли він має клікове покриття, що задовільняє умові Хеллі.*

Доведення. ⇒ Нехай маємо граф $G \simeq K(H)$, та \mathcal{E} – клікове покриття графа H , що складається із клік ($G \simeq \Omega(\mathcal{E})$). Тоді за Твердженням 2.46 $\mathcal{F} = \mathcal{F}(\mathcal{E})$ – клікове покриття графа G , за Лемою 2.48 воно задовільняє умові Хеллі.

⇐ Нехай граф G має клікове покриття $\mathcal{E} = \{Q_1, \dots, Q_n\}$, що задовільняє умові Хеллі. Із Твердження 2.46 $\mathcal{F} = \mathcal{F}(\mathcal{E}) = \{F_1, \dots, F_m\}$ є кліковим покриттям графа перетинів $H \simeq \Omega(\mathcal{E})$. Також зауважимо, що $G \simeq \Omega(\mathcal{F})$, за означенням \mathcal{F} .

Визначимо граф H^* таким чином: $V(H^*) = V(H) \cup \mathcal{F}$, $E(H^*) = E(H) \cup \{Q_i F_j : i \in F_j\}$. Кожна $F_i \in \mathcal{F}$ є вершиною графа H^* , що лежить в єдиній кліці графа H^* – $\{Q_j : j \in F_i\} \cup \{F_i\}$ (завдяки умові Хеллі). Оскільки \mathcal{F} містить кожну кліку графа H (за Лемою 2.48), то кожна кліка H^* міститиме єдину вершину $F_i \in \mathcal{F}$. Звідси $\Omega(\mathcal{F}) \simeq K(H^*)$. А оскільки $G \simeq \Omega(\mathcal{F})$, то він справді є графом клік. □

2.5 Хордальні графи

2.5.1 Означення

Означення 2.54. *Хордою* циклу називається ребро, що з'єднує дві несуміжні в циклі вершини.

Означення 2.55. Граф є *хордальним*, якщо будь-який цикл достатньої довжини, щоб мати хорду, має хорду, інакше кажучи, не містить породжених циклів довжин більших за 3.

Приклад 2.56. Приклад хордального графа з рисунком.

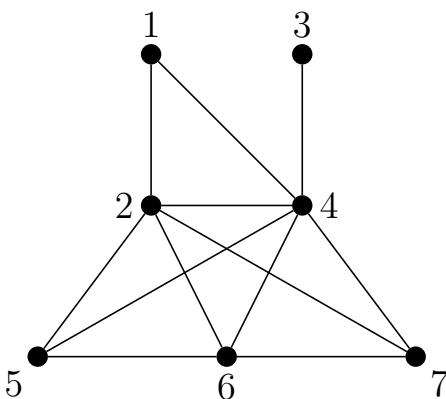


Рис. 2.20. Хордальний граф

Твердження 2.57. *Кожен граф G є графом перетинів родини підграфів деякого графа.*

Доведення. Нехай маємо граф G , $\mathcal{E} = \{Q_1, \dots, Q_m\}$ – його довільне клікове покриття та $\mathcal{F} = \mathcal{F}(\mathcal{E}) = \{S_1, \dots, S_n\}$ – дуальне представлення. Граф $G \simeq \Omega(\mathcal{F})$ – за Наслідком 2.45. Визначимо граф $H: V(H) = \{1, \dots, m\}$, $ij \in E(H)$ тоді й тільки тоді, коли $\exists k : \{i, j\} \subset S_k \in \mathcal{F}$. За Твердженням 2.46 кожен елемент \mathcal{F} містить повний підграф графа H . А за Наслідком 2.45 граф G є графом перетинів родини підграфів \mathcal{F} графа H . \square

Означення 2.58. Якщо граф G є графом перетинів підграфів графа H , то говорять, що G може бути представленим на H . Представленням $Im(u)$ вершини u графа G називається відповідний їй підграф графа H .

Лема 2.59. *Якщо граф G – цикл довжини більшої за 3, то G не може бути представлений на дереві.*

Доведення. Від супротивного, припустимо $G \simeq C_n$, $n \geq 4$ та існує дерево T , на якому G може бути представлений. Нехай $V(G) = \{v_1, \dots, v_n\}$, $E(G) = \{v_i v_j :$

$j = i \pm 1$ або $j = \pm(k - 1)$. Позначимо за $\{T_1, \dots, T_n\}$ – підграфи дерева T , $T_i \in$ представленням вершини v_i . Спершу зауважимо, що для того щоб забезпечити ребра $v_i v_{i+1}$, $i = \overline{1, k-1}$ перетин $V(T_i) \cap V(T_{i+1})$ має бути непорожній.

Якщо $V(T_i) \cap V(T_j) \neq \emptyset$, то $\forall u \in T_i, \forall w \in T_j$ існує вершина v , яка лежить на шляху між вершинами u, w та потрапляє в перетин $V(T_i) \cap V(T_j)$. Це впливає з того, що піддерева T_i, T_j зв'язні, а для будь-яких двох вершини в дереві існує єдиний шлях, що їх сполучає. Таким чином у нашій конструкції, шлях між двома вершинами $p \in T_1$ та $q \in T_n$ міститиме по вершині з кожного перетину $V(T_i) \cap V(T_{i+1}), i = \overline{1, k-1}$. А оскільки піддерево $T_n \in$ зв'язним і має містити по вершині із піддерев T_1 та T_{k-1} , то воно також міститиме і увесь шлях між цими вершинами, а отже і вершини з перетинів, а тому відповідна вершина v_n у графі G буде суміжною зі всіма іншими вершинами, що суперечиться умові $G \simeq C_n, n \geq 4$. \square

Лема 2.60. *Якщо зв'язний граф G може бути представлений на дереві T та H – повний підграф графа G , то перетин представлень вершин графа H не порожній.*

Доведення. За індукцією:

База індукції: Нехай $H \simeq K_3$, $V(H) = \{v_1, v_2, v_3\}$, а T_1, T_2, T_3 відповідні представлення вершин. Тоді $V(T_1) \cap V(T_2) \neq \emptyset$ тому, що $v_1 v_2 \in E(K_3)$. Оскільки $v_1 v_3, v_2 v_3 \in E(K_3)$, то $V(T_1) \cap V(T_3) \neq \emptyset$ та $V(T_2) \cap V(T_3) \neq \emptyset$, а отже піддерево T_3 містить вершини з дерев T_1 та T_2 . А оскільки, підграф $T_3 \in$ зв'язним, то міститиме принаймні одну вершину із множини $V(T_1) \cap V(T_2)$ (оскільки в дереві між кожною парою вершин існує єдиний шлях, то для будь-якої пари вершин $w \in T_1, u \in T_2$ шлях між ними міститиме вершину із перетину $V(T_1) \cap V(T_2)$). Таким чином, ми довели, що для C_3 теорема виконується.

Припущення індукції: нехай теорема виконується для всіх повний підграфів $C_n, n \leq p - 1$.

Крок індукції: покажемо, що для K_p теорема виконується. Розглянемо 3 повні породжені підграфи графа $K_p : K_1, K_2, K_3$. Визначимо їх таким чином: $V(K_i) = V(K_p) \setminus \{i\}$. За припущенням індукції перетин представлень вершин для кожного з графів $K_1, K_2, K_3 \in$ непорожнім. Позначимо відповідно $Q_i = \cap \{Im(u) : u \in K_i\}$. Якщо $\exists i : Im(i) \cap Q_i \neq \emptyset$, то теорема доведена, оскільки відповідний непорожній перетин $Im(i) \cap Q_i$ і є перетином представлень вершин графа. Тепер розглянемо випадок, коли $\forall i : Im(i) \cap Q_i = \emptyset$. Зафіксуємо три вершини $v_1 \in Im(1), v_2 \in Im(2), v_3 \in Im(3)$ Розглянемо три попарні шляхи $P_{v_1 v_2}, P_{v_2 v_3}, P_{v_3 v_1}$. Оскільки кожен зі шляхів знаходить на дереві, то вони мають принаймні одну спільну вершину $s \in V(P_{v_1 v_2}) \cap V(P_{v_2 v_3}) \cap V(P_{v_3 v_1})$ (інакше у дереві був би цикл). Зауважимо, що кожна з трьох множин Q_1, Q_2, Q_3 міститься одразу в усіх представленнях вершин $Im(i), i = \overline{1, p}$. $Q_2 \cup Q_3 \subseteq Im(1)$ – за побудовою, $P_{v_2 v_3} \in Im(1)$, бо піддерево $Im(1)$

зв'язне $\Rightarrow s \in Im(1)$. Аналогічно $s \in Im(2)$, $s \in Im(3)$. А отже, $s \in \cap\{Im(i) : i \in V(K_p)\}$, що закінчує доведення теореми. \square

Означення 2.61. Вершина a графа G називається *симплиціальною*, якщо породжений підграф множиною вершин $N(a) \cup \{a\}$ є повним.

Теорема 2.62. *Кожний хордальний граф має симплиціальну вершину[6].*

2.5.2 Характеризація

Теорема 2.63. *Граф є хордальним тоді і тільки тоді, коли він є графом перетинів піддерев у дереві.*

Доведення. \Rightarrow Нехай існує не хордальний граф G , що може бути представлений на дереві T . Тоді G містить породжений цикл C довжини принаймні 4. Оскільки C зв'язний, то об'єднання представлень його вершин є піддерево $T' \subseteq T$. Але тоді C може бути представлений на дереві T' , що суперечить Лемі 2.59.

\Leftarrow За індукцією

База індукції: Якщо хордальний граф G має одну точку, то G може бути тривіально представленим на дереві.

Припущення індукції: нехай будь-який хордальний граф G з $|V(G)| < n$ може бути представлений на дереві.

Крок індукції: нехай граф G має n вершин. За Теоремою 2.62 він має симплиціальну вершину v_0 . Розглянемо граф $G - v_0$, він зв'язний тому, що кожні дві вершини суміжні із видаленою вершиною v_0 є суміжними між собою. І також він є хордальним тому, що будь-який цикл графа $G - v_0$ є циклом хордального графа G . Таким чином за припущенням індукції граф $G - v_0$ може бути представленим на дереві.

Оскільки v_0 симплиціальна вершина, то $N(v_0)$ є повним підграфом, тоді за Лемою 2.60 перетин представлень вершин із множини $N(v_0)$ є непорожнім. Нехай вершина u_0 лежить у цьому перетині.

Нехай P – ланцюг на 3 вершинах: $V(P) = \{u_1, u_2, u_3\}$, де u_1, u_3 – його кінці. Визначимо дерево T' приклеївши ланцюг P до дерева T по вершинах $u_0 = u_1$. Тепер ми можемо представити граф G на дереві T' таким чином. Для вершин $u_i \in G - N(v_0) - v_0$ представлення зберігається. Тобто, якщо її представленням було піддерево T_i дерева T , то її і представлятиме те ж піддерево T_i дерева T' . Якщо $u_i \in N(v_0)$ та T_i представляє u_i на дереві T , то тоді $T_i \cup P$ буде її представленням на дереві T' . Для вершини v_0 представленням буде шлях u_2u_3 із приклеєного ланцюга P . Таким чином не утворюється жодних нових ребер. Представлення v_0 перетинається лише з представленнями вершин із $N(v_0)$. \square

Висновки

В нашій роботі проведено дослідження теми графів перетинів. Основна мета – дослідження класів графів перетинів є досягнутою. Для кожного з них були надані характеристики, деякі результати ілюструються малюнками. Також додається код, що ілюструє деякі отримані результати. На мові *Haskell* наводяться алгоритми побудови реберного графа. А на мові *C++* наведені алгоритми знаходження точок зв'язності, двозв'язних компонент, побудови графа блоків та знаходження чисел $\beta_0, \beta_1, \beta_2$ для довільного зв'язного графа.

Основними висновками по розділах роботи є: - в **Розділі 1** надані базові означення, що використовуються далі в роботі. Зокрема вводиться поняття графа перетинів.

Розділ 2 – це основна частина роботи. У **Підрозділі 2.1** наводяться попередні результати. Зокрема доводиться, що будь-який граф можна представити як граф перетинів.

У **Підрозділі 2.2** наведено означення графа блоків, як графа перетинів на множині блоків. Також надана характеристика: для графа H існує граф G такий, що $H \simeq B(G)$ тоді й тільки тоді, коли кожен блок H є повним підграфом. Далі розглянуто блоковий оператор. Зокрема розглядається ітерація блокового оператора. Було доведено, що мінімальне число n таке, що $B^n(G) \simeq K_1$ це $\text{diam}(G)$, що є власним результатом.

У **Підрозділі 2.3** розглянуто реберні графи. Зокрема доведено, що $B(G) \simeq L(G)$ тоді й тільки тоді, коли граф G є деревом. Надана характеристика реберних графів, як графів множини вершин яких можна розбити на повні підграфи так, щоб кожна вершина належала не більше ніж двом цим підграфам.

У **Підрозділі 2.4** визначаються графи клік. Надається характеристика, як граф, що має клікове покриття, яке задовільняє умові Хеллі. Також отримано критерій, коли для графа G існує двочастковий граф H , такий що $L(H) \simeq G$. В собі доведення використовує оператор клік та алгоритм із характеристики реберних графів.

У **Підрозділі 2.5** розглянуто клас хордальних графів. Доведено Лему про неможливість представлення породжених циклів C_n , $n \geq 4$ на дереві, та Лему 2.52, яка говорить, що для графа представленого на дереві, перетин представлень вершин будь-якого його повного підграфа непорожній. Також, надана характеристика, як графів перетинів піддерев у дереві, яка в доведенні використовує дві вищеперераховані леми.

Отже, ця тема має не тільки наукову новизну, але і є темою для глибошого вивчення і дослідження.

Література

- [1] Lowell W. Beineke, *Characterizations of Derived Graphs*, Journal of Combinatorial Theory, **9** (1970), 129–135.
- [2] P. Erdos, C. Ko, R. Rado, *Intersection theorems for systems of finite sets*, Quarterly Journal of Mathematics Oxford Ser. II **12** (1961) 313–318.
- [3] Frank Harary (1969), *Graph Theory*, Addison-Wesley Publishing Company, Inc.
- [4] Frank Harary, *A Characterization of Block-Graphs*, Canadian Mathematical Bulletin, **6(1)** (1963), 1–6.
- [5] John Hopcroft and Robert Tarjan, *Efficient Algorithms for Graph Manipulation /H/*, Communications of the ACM **16(6)** (1973), 372–378.
- [6] C. Lekkerkerker and J. Boland, *Representation of a finite graph by a set of intervals on the real line*, Fundamenta Mathematicae, **51** (1962), 45–64.
- [7] Singh Anurag, *Biconnected Components*, 25 Січ. 2020 [Електронний ресурс] <https://www.geeksforgeeks.org/biconnected-components/>
- [8] Terry A. McKee F. R. McMorris (1999), *Topics in intersection graph theory*, Society for Industrial and Applied Mathematics.
- [9] James R. Walter, *Representations of Chordal Graphs as Subtrees of -a Tree*, Journal of Graph Theory **2** (1978), 265–267.
- [10] *Articulation Points (or Cut Vertices) in a Graph*, 25 Бер. 2021 [Електронний ресурс] <https://www.geeksforgeeks.org/articulation-points-or-cut-vertices-in-a-graph/>