

Міністерство освіти і науки України
Національний університет «Києво-Могилянська академія»
Факультет інформатики
Кафедра математики

Кваліфікаційна робота
освітній ступінь — бакалавр

на тему: «**ЗАДАЧА ДІНИЦЯ**»

Виконала: студентка 4-го року
навчання
освітньої програми «Прикладна
математика»
спеціальності 113 Прикладна
математика

Сапіга Ольга Юріївна

Керівник: Тимошкевич Л.М.,
кандидат фіз.-мат. наук, ст. викладач

Рецензент _____
(прізвище та ініціали)

Кваліфікаційна робота захищена
з оцінкою _____

Секретар ЕК _____

«_____» _____ 20____ р.

Графік підготовки кваліфікаційної роботи до захисту

Графік узгоджено “_____” _____ 2021 р.

№ з/п	Перелік робіт	Термін виконання етапу	Підпис наукового керівника	Дата ознайомлення наукового керівника	Примітка
1.	Отримання теми кваліфікаційної роботи.	09.10.2020			
2.	Ознайомлення з темою кваліфікаційної роботи.	11.10.2020			
3.	Розробка плану та структури роботи.	22.11.2020			
4.	Робота з науковою літературою, опис основних означень.	29.12.2020			
5.	Дослідження результатів отриманих в літературі.	05.02.2021			
7.	Робота над текстовим оформленням результатів.	18.03.2021			
8.	Попередній аналіз кваліфікаційної роботи. Виправлення помилок.	28.04.2021			
9.	Попередній захист кваліфікаційної роботи.	14.05.2021			
10.	Захист кваліфікаційної роботи.	09.06.2021			

Науковий керівник _____

(ПІБ)

Виконавець кваліфікаційної роботи _____

(ПІБ)

Зміст

1	Вступ	3
1.1	Актуальність	3
1.2	Мета, завдання дослідження	4
2	Задача Діниця	5
2.1	Постановка задачі Діниця	5
2.2	Латинські квадрати	5
2.2.1	Основні означення про латинські квадрати	6
2.2.2	Представлення латинського квадрата	8
2.2.3	Число латинських квадратів	9
2.3	Загальний випадок задачі Діниця	10
2.4	Формулювання задачі Діниця мовою теорії графів	10
2.4.1	Розфарбування за списком	11
2.5	$\chi_l(S_n) \leq n + 1$	17
2.5.1	Теорема 1. Орієнтовані графи	17
2.5.2	Теорема Холла. Паросполучення у двочастковому графі	20
2.5.3	Теорема 2. Стійкі паросполучення	24
2.6	$\chi_l(S_n) = n$	30
2.7	Практична частина	32
2.7.1	Алгоритм знаходження розфарбування для задачі Діниця	32
2.7.2	Задача 1.1	35
2.7.3	Задача 1.2	38
2.7.4	Задача 2.1	43
2.7.5	Задача 2.2	44
2.8	Гіпотеза про хроматичні числа реберних графів	46
2.9	Прикладне застосування	48
3	Висновки	52
4	Список літератури	54

1 Вступ

1.1 Актуальність

Сучасна математика — наука, яка охоплює багато розділів, потрібно дуже багато часу, щоб ознайомитися з ними. Так виділяють задачі, які є надважливими, бо вони є відправними точками для дослідження і розвитку розділу, в якому вони з'явилися. Одна з таких задач — задача про хроматичне число. Ця задача є важливою і популярною, відноситься до розділу комбінаторної геометрії і теорії графів. Комбінаторна геометрія — наука про комбінаторні властивості геометричних об'єктів. Теорія графів — наука, що динамічно розвивається, це розділ дискретної математики, що вивчає властивості графів.

Гіпотеза Діниця, яка тепер є теоремою Діниця, була відкритою проблемою щодо розфарбування графа чи масиву, що поєднує елементи комбінаторики та теорії графів. Задача була запропонована Джеффом Діницем у 1979 році і лише у 1994 році Фред Гальвін знайшов розв'язання.

Доведення Гальвіна узагальнює твердження, що для кожного двочасткового графа хроматичне число за списком дорівнює його хроматичному числу. Загальна гіпотеза стверджує, що те саме справедливо не лише для двочасткових графів, але і для будь-якого графа без петель.

Задача Діниця є популярною і породила іншу гіпотезу, яка досі залишається нерозв'язаною: чи для будь-якого реберного графа виконується, що хроматичне число дорівнює хроматичному числу за списком.

Розфарбування та розфарбування списком графів є важливими і мають багато застосувань, а саме в: створенні розкладів, розподілі регістрів в мікропроцесах, розподілі частот, розпаралелюванні чисельних методів, обчисленні похідних, алгоритмі для цифрових водяних знаків, кластерному аналізі та розв'язуванні sudoku.

1.2 Мета, завдання дослідження

1. Розглянути постановку задачі Діниці і формулювання задачі Діниці мовою теорії графів.
2. Зробити огляд основних означень та тверджень, що пов'язані з теорією розфарбування графів, які використовуються в задачі Діниці.
3. Продемонструвати на прикладах застосування методів та понять, що використовуються в доведенні теорем і тверджень.
4. Записати структуру алгоритму для розв'язання задачі Діниці в загальному випадку і застосувати його до часткових прикладів.
5. Розглянути постановку алгоритму Гейла-Шеплі з доведенням і застосування його на часткових прикладах.

2 Задача Діниця

2.1 Постановка задачі Діниця

Розглянемо n^2 клітин, які утворюють квадрат розміру $n \times n$. Нехай (i, j) позначає клітину на перетині рядка i і стовпця j . Для кожної клітини (i, j) задана множина, що позначимо $C(i, j)$, яка складається з n різних кольорів.

Питання задачі полягає в тому, що чи завжди можна розфарбувати всі клітини квадрата так, щоб кольори клітин в кожному рядку і кожному стовпцю були різними. Кольори для кожної клітини (i, j) обираємо з множини $C(i, j)$.

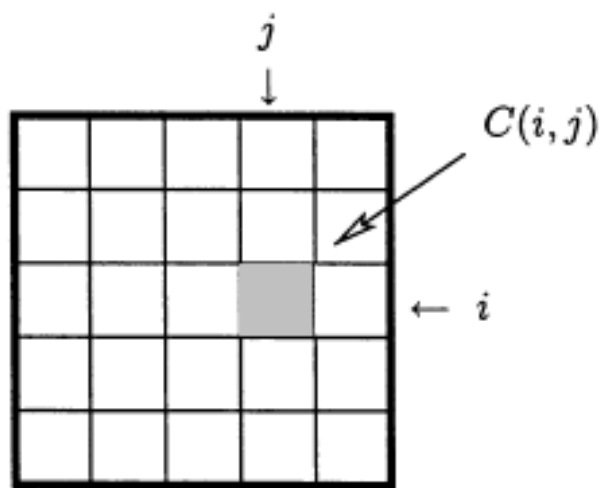


Рис. 1 [1]

Якщо всі множини $C(i, j)$ однакові, наприклад, маємо, що $C(i, j) = \{1, 2, \dots, n\}$ для всіх i, j . Тоді задача Діниця зводиться до задачі: потрібно заповнити клітини квадрата розміру $n \times n$ числами $1, 2, \dots, n$ так, щоб числа в клітинах кожного рядка і стовпця були різними. Таке розфарбування є **латинським квадратом**.

2.2 Латинські квадрати

Латинські квадрати є одними з найстаріших комбінаторних об'єктів, які вивчали ще в античні часи.

2.2.1 Основні означення про латинські квадрати

Означення 1. *Латинський квадрат n -го порядку* — це таблиця $L = (l_{ij})$ розміру $n \times n$, що заповнена n елементами з множини M таким чином, що в кожному рядку і кожному стовпцю таблиці кожний елемент з множини M зустрічається тільки один раз. Тобто рядки і стовпці є перестановками елементів множини M .

Множина M — це зазвичай множина, що складається з множини натуральних чисел $\{1, 2, \dots, n\}$ або множини $\{0, 1, \dots, n-1\}$. Ейлер використовував букви латинського алфавіту, тому латинські квадрати і отримали таку назву.

Латинські квадрати існують для будь-якого n . Можна взяти таблицю Келі групи порядку n , наприклад $l_{ij} = (i + j - 1) \bmod n$.

Означення 2. *Два латинських квадрата $L = (l_{ij})$ і $K = (k_{ij})$ n -го порядку називаються ортогональними*, якщо всі впорядковані пари (l_{ij}, k_{ij}) є різними.

1	2	3	4
2	1	4	3
4	3	1	2
3	4	2	1

Ейлер називав такі квадрати **повними**. Повний латинський квадрат порядку 4 можна побачити на рис.2. Приклад трьох ортогональних латинських квадратів і відповідні їм впорядковані пари можна побачити на рис.3.

Рис. 2 [1]

1	2	3	4
2	3	4	1
4	1	2	3
3	4	1	2

1	2	3	4
3	4	1	2
2	3	4	1
4	1	2	3

1	2	3	4
4	1	2	3
3	4	1	2
2	3	4	1

(1,1,1)	(2,2,2)	(3,3,3)	(4,4,4)
(2,3,4)	(3,4,1)	(4,1,2)	(1,2,3)
(4,2,3)	(1,3,4)	(2,4,1)	(3,1,2)
(3,4,2)	(4,1,3)	(1,2,4)	(2,3,1)

Рис. 3

Означення 3. *Неповний латинський квадрат порядку n* — це таке заповнення клітин числами з множини $\{1, 2, \dots, n\}$ квадрата $n \times n$, що кожне

число зустрічається не більше одного разу в кожному рядку і стовпцю.

Приклади доповнення неповних латинських квадратів для утворення повних латинських квадратів того ж порядку:

1) Нехай перші $n - 1$ рядків заповнені, а останній рядок пустий. В цьому неповному латинському квадраті кожне число зустрічається $n - 1$ раз. Тобто кожному стовпцю не вистачає одного числа. Якщо запишемо кожне число в нижній рядок і в тому стовпцю, де немає цього числа, то отримаємо повний латинський квадрат.

1	4	2	5	3
4	2	5	3	1
2	5	3	1	4
5	3	1	4	2
3	1	4	2	5

Рис. 4 [1]

2) Нехай заповнений тільки перший рядок, а всі інші пусті. Тоді можна заповнити кожний наступний рядок циклічним зсувом вправо або вліво на один крок і отримати повний латинський квадрат. Приклад циклічного латинського квадрата порядку 5 можна побачити на рис.4.

Приклад неповного латинського квадрата, що неможливо доповнити до повного латинського квадрата:

Неповний латинський квадрат, що можна побачити на рис.5, неможливо доповнити до повного латинського квадрата. Заповнені тільки n клітин, але не в одному рядку, тому верхню праву клітину неможливо заповнити числом, яке не буде повторенням числа, що вже є в рядку або стовпці.

1	2	...	$n-1$	
				n

Рис. 5 [1]

2.2.2 Представлення латинського квадрата

Означення 4. Латинський квадрат $n \times n$ розміру можна представити у вигляді таблиці розміру $n \times n^2$, яка називається *лінійною таблицею латинського квадрата*.

Розглянемо наприклад латинський квадрат розміру 3×3 . Йому відповідає лінійна таблиця розміру 3×9 . Латинський квадрат порядку 3 і відповідну йому таблицю можна побачити на рис.6.

1	3	2
2	1	3
3	2	1

R: 1 1 1 2 2 2 3 3 3

C: 1 2 3 1 2 3 1 2 3

E: 1 3 2 2 1 3 3 2 1

R позначає рядок квадрата, **C** позначає стовець, а **E** — елементи латинського квадрата. Еквівалентність латинського квадрата і таблиці полягає в тому, що умова для утворення латинського квадрата еквівалентна тому, що в кожних двох рядках лінійної таблиці буде n^2 упорядкованих пар елементів з множини $\{1, 2, \dots, n\}$. Кожна пара буде зустрічатися тільки один раз.

Рис. 6 [1]

Означення 5. Латинські квадрати називаються *спряженими*, якщо вони пов'язані перестановкою рядків в лінійній таблиці.

Приклад спряженого латинського квадрата до латинського квадрата, що на рис.6:

1	2	3
3	1	2
2	3	1

R: 1 3 2 2 1 3 3 2 1

C: 1 1 1 2 2 2 3 3 3

E: 1 2 3 1 2 3 1 2 3

Якщо взяти лінійну таблицю з рис.6 і переставити циклічно рядки $R \rightarrow C \rightarrow E \rightarrow R$, то можна отримати латинський квадрат, який можна побачити на рис.7 і він буде спряженим до латинського квадрата, що на рис.6.

Рис. 7 [1]

Властивості неповних латинських квадратів:

- Неповний латинський квадрат відповідає неповній лінійній таблиці, так як в будь-яких двох рядках таблиці кожна пара елементів з множини $\{1, 2, \dots, n\}$ зустрічається не більше одного разу.
- Латинський квадрат, який є спряженим до неповного латинського квадрата, теж буде неповним латинським квадратом.
- Неповний латинський квадрат можна доповнити до повного тоді і тільки тоді, коли будь-який спряжений до неповного можна доповнити до повного.

2.2.3 Число латинських квадратів

Формула для підрахунку числа $L(n)$ латинських квадратів порядку n досі не знайдена, але найкращу оцінку числа $L(n)$ дає формула:

$$\prod_{k=1}^n (k!)^{n/k} \leq L(n) \leq \frac{(n!)^{2n}}{n^{n^2}}$$

Точні значення для числа $L(n)$ відомі тільки для n від 1 до 11:

n	$L(n)$
1	1
2	2
3	12
4	576
5	161280
6	812851200
7	61479419904000
8	108776032459082956800
9	5524751496156892842531225600
10	9982437658213039871725064756920320000
11	776966836171770144107444346734230682311065600000

Табл. 1 [12]

2.3 Загальний випадок задачі Діниця

З латинськими квадратами цю задачу можна легко розв'язати. Задача складнішає у випадку, коли кольорів більше ніж n в множині C . Тобто множина

$$C := \bigcup_{i,j} C(i, j)$$

В цьому випадку для розфарбування клітини квадрата не можна брати будь-який колір з множини C .

{1, 2}	{2, 3}
{1, 3}	{2, 3}

Рис. 8 [1]

Приклад:

Розглянемо квадрат розміру 2×2 і множина $C = \{1, 2, 3\}$ (рис.8). Якщо обрати для розфарбування першого рядка кольори 1 і 2, то дві клітини другого рядка потрібно буде розфарбувати в колір 3, і вже виникає суперечність умові задачі Діниця.

2.4 Формулювання задачі Діниця мовою теорії графів

Означення 1. Графом $G(V, E)$ називається сукупність двох множин: не пустої множини V (множина вершин) і множини E (множина ребер), що є двохелементною підмножиною множини V .

$$G(V, E) = \langle V; E \rangle, V \neq \emptyset, E \subset 2^V$$

Означення 2. Дистанційний граф або граф одиничних відрізків на площині — будь-яка конфігурація, вершини якої точки площини, а ребра — відрізки довжини 1, які з'єднують пари точок, що знаходяться на відстані 1.

Не обов'язково з'єднувати кожну пару вершин ребром. Граф буде дистанційним, навіть якщо буде складатися зі скінченної множини вершин. Позначення для дистанційного графа — $G = (V, E)$, де V — це множина вершин(скінченна), а E — множина ребер(теж скінченна).

Означення 3. Хроматичне число графа G — мінімальне число кольорів $\chi(G)$, в які можна так розфарбувати вершини графа, щоб вершини будь-

якого ребра мали відмінний колір.

Тобто таке розфарбування розбиває множину вершин графа на класи, що складаються з вершин одного кольору. Між вершинами, що знаходяться в одному класі, ребер немає.

Означення 4. Нехай $A \subseteq V$, тоді множина A називається *незалежною*, якщо вершини з множини A не пов'язані між собою ребрами графа G . Тоді хроматичне число — це найменше число незалежних множин, що породжують розбиття множини вершин V .

2.4.1 Розфарбування за списком

У 1976 році Візінг В.Г. і через 3 роки Ердеш, Рубін та Тейлор розглядали такий варіант розфарбування, що приводить до розв'язання задачі Діниця.

Нехай для кожної вершини v графа $G = (V, E)$ задана множина кольорів $C(v)$.

Означення 5. *Розфарбування за списком* — це таке відображення $c : V \rightarrow \bigcup_{v \in V} C(v)$, що $c(v) \in C(v)$ для кожного $v \in V$.

Означення 6. *Хроматичне число розфарбування за списком* $\chi_l(G)$ — це таке найменше число k , що для будь-якого списку множини кольорів $C(v)$ з кількістю кольорів $|C(v)| = k$, буде існувати розфарбування за списком для кожного $v \in V$.

За умовою $\chi_l(G)$ — це таке найменше число k , тому можна зробити висновки, що $\chi_l(G)$ буде не перевищувати потужності вершин графа G , тобто $\chi_l(G) \leq |V|$. Кожній вершині v , що належить множині вершин графа G , ставиться у відповідність множина кольорів $C(v)$ і, якщо потужність множини кольорів буде більше за потужність множини вершин V , то з'явиться надмірність у виборі кольору з множини $C(v)$ для кожної вершини v , що суперечить

означенню хроматичного числа.

Твердження 1. $\chi(G) \leq \chi_l(G)$.

Доведення.

За означенням хроматичного числа графа $\chi(G)$ — мінімальне число кольорів для розфарбування кожної вершини $v \in V$ (множина кольорів $C(v)$ однакова для всіх вершин), щоб вершини одного ребра мали відмінний колір. Такий випадок розфарбування є частковим для розфарбування за списком, яке може мати різні набори кольорів множин $C(v)$ для кожної вершини $v \in V$. Нехай $\chi_l(G) = k$, тобто різні множини $C(v)$ з деякого списку мають потужність k .

$\chi(G) = \chi_l(G)$, коли список має однакові множини $C(v)$ потужності k .

$\chi(G) < \chi_l(G)$, коли хоча б одна множина зі списку потужності k буде відрізнятися набором кольорів від інших множин $C(v)$ потужності k .

Отже, $\chi(G) \leq \chi_l(G)$. □

Означення 7. Граф S_n — граф, множина вершин якого складається з n^2 клітин, які утворюють таблицю розміру $n \times n$, і дві вершини цього графа суміжні тоді і тільки тоді, коли знаходяться в одному і тому ж рядку і стовпцю. Приклад графа S_3 на рис.9.

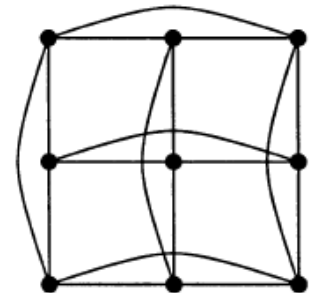


Рис. 9 [1]

Твердження 2. Хроматичне число графу S_n буде дорівнювати n , тобто $\chi(S_n) = n$.

Доведення.

Розглянемо наприклад перший рядок таблиці розміру $n \times n$, що утворює граф S_n . Пофарбуємо цей рядок в $n - 1$ колір. Отримаємо суперечність з означенням хроматичного числа, так як за принципом Діріхле знайдуться дві вершини, які будуть пофарбовані в один колір і також будуть суміжни-

ми, так як знаходяться в одному рядку(рис.10). Отже, для розфарбування тільки одного рядка потрібно щонайменше n кольорів.



Рис. 10

Для стовпця також отримаємо, що для розфарбування потрібно $\geq n$ кольорів. Отже, для розфарбування графа S_n потрібно не менше n кольорів.

За означенням хроматичного числа, це мінімальне число кольорів для розфарбування графа, тому $\chi(S_n) = n$. $\square$

Для графа S_n , так як і для будь-якого графа, буде існувати розфарбування за списком, і тому справедлива нерівність $\chi(S_n) \leq \chi_l(S_n)$. **Задачу Діниці** можна переформулювати таким чином: чи існує розфарбування за списком графа S_n , яке буде дорівнювати n , тобто $\chi_l(S_n) = n$?

Означення 8. Граф $G = (V, E)$ називається двочастковим, якщо множини його вершин можна розбити на дві частини $V = U \cup W$ так, що жодна вершина з підмножини U не з'єднана з іншими вершинами в підмножині U і жодна вершина з підмножини W не з'єднана з іншими вершинами в підмножині W . Підмножини вершин U і W називаються *частками* двочасткового графа G .

Означення 9. Повним двочастковим графом називається двочастковий граф, в якому для кожної пари вершин $u \in U$, $w \in W$ існує ребро $(u, v) \in E$. Такий граф позначається $K_{i,j}$, де $|U| = i$, $|W| = j$.

Приклад того, що рівність $\chi(G) = \chi_l(G)$ не справджується для графа $K_{2,4}$:

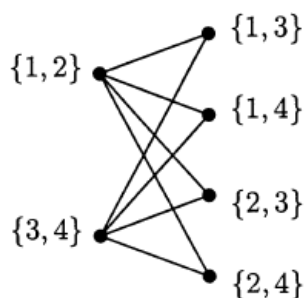


Рис. 11 [1]

Розглянемо граф $G = K_{2,4}$. Хроматичне число будь-якого двочасткового графа дорівнює 2, так як вершини однієї частки можна розфарбувати в один колір, а вершини другої частки в інший колір. Тому $\chi(K_{2,4}) = 2$.

Тепер знайдемо хроматичне число цього графа для розфарбування за списком. Нехай задані такі множини кольорів, які показані на рис.11.

Розглянемо **4 варіанти розфарбування лівої частки**:

1) Першу вершину фарбуємо в колір 1, другу в колір 3. Тоді для розфарбування першої вершини в правій частці графа не залишилося кольорів, а отже правильного розфарбування для такого варіанту не існує.

2) Першу вершину фарбуємо в колір 1, другу в колір 4. Тоді для розфарбування другої вершини в правій частці графа не залишилося кольорів, а отже правильного розфарбування для такого варіанту не існує.

3) Першу вершину фарбуємо в колір 2, другу в колір 3. Тоді для розфарбування третьої вершини в правій частці графа не залишилося кольорів, а отже правильного розфарбування для такого варіанту не існує.

4) Першу вершину фарбуємо в колір 2, другу в колір 4. Тоді для розфарбування четвертої вершини в правій частці графа не залишилося кольорів, а отже правильного розфарбування для такого варіанту не існує.

Отже, можна зробити висновок, що $\chi_l(K_{2,4}) \geq 3$, що не дорівнює хроматичному числу $\chi(K_{2,4}) = 2$.

Твердження 3. Для графа $K_{2,4}$ хроматичне число розфарбування за списком буде дорівнювати трьом, тобто $\chi_l(K_{2,4}) = 3$.

Доведення.

Раніше ми побачили, що у графа $K_{2,4}$ хроматичне число розфарбування за списком дорівнює щонайменше три. Тепер доведемо, що $\chi_l(K_{2,4}) = 3$.

Нехай задані деякі множини кольорів для кожної вершини, що складаються вже з трьох різних кольорів. Розглянемо дві вершини з лівої частки і будь-яку з точок правої долі графа. Так як маємо три різні кольори для вибору з множини кольорів, то незважаючи на те, які кольори оберемо для розфарбування вершин лівої частки, для розфарбування вершини з правої частки завжди знайдеться хоча б один колір з множини кольорів, щоб розфарбування суміжних вершин було правильним.

Отже, $\chi_l(K_{2,4}) = 3$. □

Приклад того, що рівність $\chi(G) = \chi_l(G)$ не справджується для графа $K_{3,27}$:

Покажемо, що насправді рівність $\chi(G) = \chi_l(G)$ не справджується ще для одного двочасткового графа $K_{3,27}$. Також доведемо, що хроматичне число такого графа буде дорівнювати щонайменше чотирьом.

Розглянемо граф $G = K_{3,27}$. Хроматичне число будь-якого двочасткового графа дорівнює 2, тому $\chi(K_{3,27}) = 2$.

Тепер знайдемо хроматичне число цього графа для розфарбування за списком. Нехай задані такі множини кольорів, які показані на рис.12. Множина кольорів для внутрішніх вершин графа буде така $\{i0, i1, i2, \}$, де перші знаки рівні для кожного вибору з трьох можливостей вибору цифри i . Для вершин ззовні множина кольорів буде така $\{0a, 1b, 2c, \dots\}$, що перша цифра буде відрізнятися для будь-якого набору (a, b, c) з трьох елементів. Таких наборів буде всього 27, що дорівнює кількості зовнішніх вершин.

Обираємо один з трьох запропонованих кольорів $\{00, 01, 02\}$ для розфар-

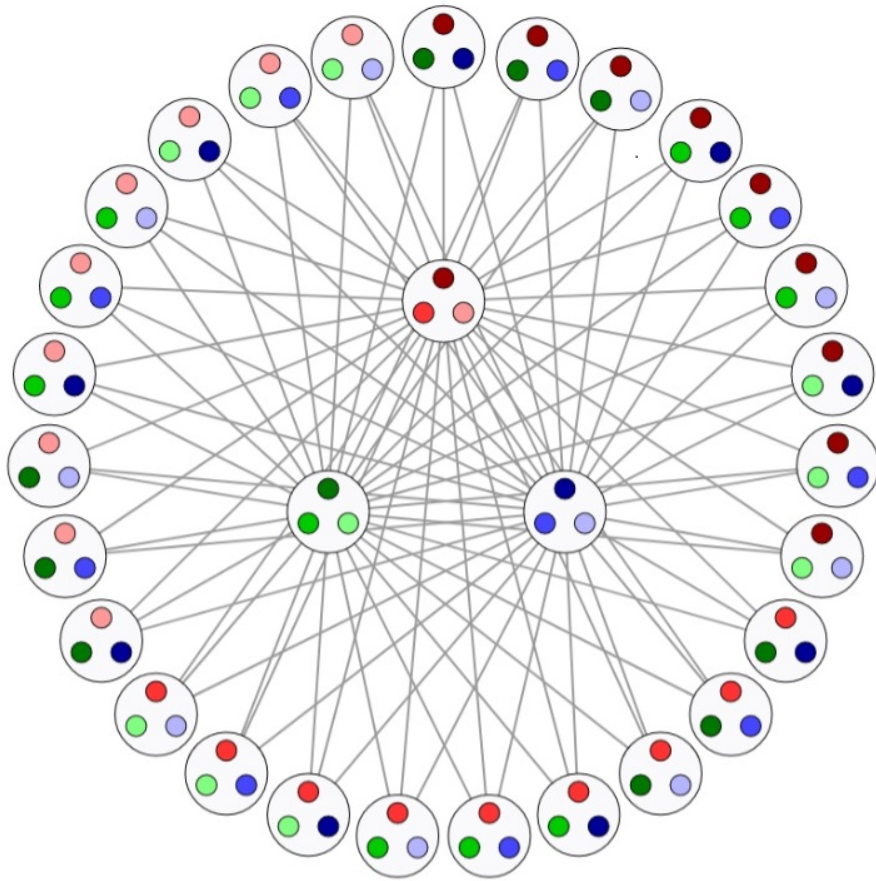


Рис. 12 [6]

бування першої внутрішньої вершини (нехай це буде вершина з множини червоних кольорів). Нехай це буде колір 00.

Обираємо один з трьох запропонованих кольорів $\{14, 15, 16\}$ для розфарбування другої внутрішньої вершини (нехай це буде вершина з множини зелених кольорів). Нехай це буде колір 14.

Обираємо один з трьох запропонованих кольорів $\{27, 28, 29\}$ для розфарбування третьої внутрішньої вершини (нехай це буде вершина з множини синіх кольорів). Нехай це буде колір 27.

Тоді отримаємо, що хоча б одна вершина буде мати множину вибору кольорів, що буде складатися з вибраних кольорів $\{00, 14, 27\}$. А отже, для цієї вершини не можна обрати колір, щоб утворилося правильне розфарбування.

Хроматичне число розфарбування за списком графа $K_{3,27}$ повинно дорівнювати щонайменше чотирьом, щоб утворилося правильне розфарбування. А отже, **хроматичне число $\chi(K_{3,27}) = 2$ не буде дорівнювати хроматичному числу розфарбування за списком $\chi_l(G) \geq 4$.**

2.5 $\chi_l(S_n) \leq n + 1$

У 1992 році Джанет Янсен довела, що $\chi_l(S_n) \leq n + 1$ і далі розв'язок завершив Фред Галвін, доповнивши двома відомими результатами.

2.5.1 Теорема 1. Орієнтовані графи

Означення 1. Нехай v — деяка вершина графа $G = (V, E)$. *Степінню вершини v* називається кількість ребер графа G , інцидентних вершині v (тобто кількість ребер, що виходить з вершини v) і позначається d_v .

Степінь кожної вершини в графі S_n дорівнює $2n - 2$. В кожному рядку кожна вершина суміжна з $n - 1$ іншими вершинами в тому ж рядку, в кожному стовпці кожна вершина суміжна з $n - 1$ іншими вершинами в тому ж стовпці. Отже, $d_v = n - 1 + n - 1 = 2n - 2$.

Означення 2. Для деякої підмножини $A \subseteq V$ позначимо G_A — підграф графа $G = (V, E)$, який має множину вершин A і множина ребер складається з усіх ребер графа G між вершинами з множинами A . Тоді G_A називається *підграфом, що індукований множиною A* . Якщо $H = G_A$ для деякої підмножини $A \subseteq V$, то будемо казати, що H — *індукований підграф* графа G .

Означення 3. *Орієнтованим графом $G = (V, E)$* називають сукупність двох множин — непорожньої множини вершин V і множини орієнтованих ребер (дуг) E . Кожне ребро з'єднує упорядовану пару вершин. Кажуть, що дуга зв'язана з упорядкованою парою вершин (u, v) , якщо ребро починається у вершині u і закінчується у вершині v . Напрям ребра позначають стрілкою. Позначають $e = (u, v)$ або $u \rightarrow v$.

Степінь виходу з вершини позначають, як d_v^+ , степінь входу у вершину позначають, як d_v^- . Тобто d_v^+ — кількість дуг з початковою вершиною v , d_v^- — кількість дуг з кінцевою вершиною v . Виконується рівність $d_v = d_v^+ + d_v^-$. G — позначення для графа $\vec{G}$ без орієнтації.

Означення 4. Множина вершин графа називається *незалежною*, якщо ніякі дві вершини цієї множини не з'єднані ребром. Іншими словами, *індукований цією множиною підграф складається з ізольованих вершин*.

Означення 5. Нехай $\vec{G} = (V, E)$ — оргграф. Ядром $K \subseteq V$ називається така підмножина V , що

- K — незалежна множина в G
- для кожної вершини $u \in V \setminus K$ існує дуга $u \rightarrow v$, $v \in K$.

Приклад:

Розглянемо оргграф $\vec{G} = (V, E)$ (рис.13). Вершини b, c, f утворюють ядро цього графу. Підграф, що індукований множиною a, c, e не має ядра, так як три дуги (a, e) , (e, c) , (c, a) утворюють цикл. Підграф, що індукований множиною d, c, f також не має ядра, так як не виконується умова друга з означення ядра: не існує дуги $v \in V \setminus K \rightarrow d$.

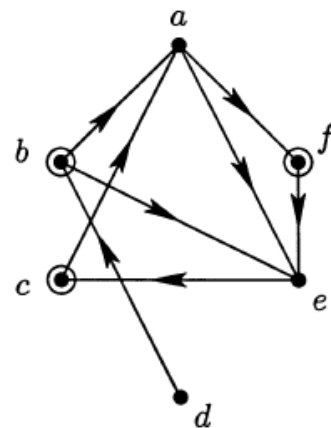


Рис. 13 [1]

Теорема 1. Нехай $\vec{G} = (V, E)$ — оргграф і для кожної вершини $v \in V$ задана множина кольорів $C(v)$. Потужність множини кольорів більша степені виходу, тобто $|C(v)| \geq d_v^+ + 1$. Якщо кожний індукований підграф графа $\vec{G}$ має ядро, то існує розфарбування за списком графа G , в якому кожна вершина $v \in V$ пофарбована в один з кольорів, що належить множині кольорів $C(v)$.

Доведення.

Доведення проведемо індукцією по $|V|$.

База: $|V| = 1$. Очевидно, що тоді $v \in V$ — ізольована вершина і степінь вершини дорівнює нулю. Тому маємо, що $|C(v)| \geq 1$. Це дійсно так, для розфарбування достатньо одного кольору.

Крок: Припустимо, що лема виконується для всіх орграфів з числом вершин меншим від $|V|$. Виберемо деякий колір $c \in C = \cup_{v \in V} C(v)$ і покладемо:

$$A(c) := \{v \in V : c \in C(v)\}$$

$A(c)$ — множина, що складається у усіх вершин графа $\vec{G}$, що мають у своєму списку кольорів для розфарбування колір c .

За умовою індукований підграф $G_{A(c)}$ має ядро $K(c)$. Пофарбуємо всі вершини $v \in K(c)$ в колір c . Це можливо зробити, так як $K(c)$ — незалежна множина. Далі видалимо множину $K(c)$ з G , а колір c видалимо з множини C . За означенням ядра для кожної вершини $u \in V \setminus K$ існує дуга $u \rightarrow v$, $v \in K$, а отже при видаленні множини $K(c)$ з G видаляються і дуги, що з'єднували ядро з графом G .

Нехай G' — індукований підграф графа G на $V \setminus K(c)$ з множиною кольорів $C'(v) = C(v) \setminus c$. Для кожної вершини $v \in A(c) \setminus K(c)$ степінь виходу $d_v'^+ \leq d_v^+ - 1$, за другою властивістю ядра степінь виходу зменшилася щонайменше на один через видалення ядра $K(c)$.

Множина кольорів $C(v)$ має потужність не більше n , як зазначалося раніше потужність множини кольорів не можуть перевищувати кількості вершин графа G , що дорівнює n . Степінь виходу кожної вершини графа G дорівнює не більше $n - 1$.

Тому виконується нерівність $d_v'^+ + 1 \leq C'(v)$. Ця нерівність справедлива і для вершин, що знаходяться не в множині $A(c)$, так як для них множина кольорів $C(v)$ не змінюється.

Новий граф G' має меншу кількість вершин, ніж граф G . За припущенням індукції для нового графу G' і для множин $C'(v)$ існує розфарбування за списком. Якщо повернути ядро $K(c)$ до множини G , то розфарбування не зміниться, а число кольорів для розфарбування графа G збільшиться на один. Індукційний крок зроблено. $\square$

Ця лема допоможе у розв'язанні **задачі Діниці**, тепер достатньо знайти орієнтацію графа S_n із степенями виходу $d_v^+ \leq n - 1$ для всіх v . Орієнтація повинна бути така, що всі індуковані підграфи будуть мати ядра.

2.5.2 Теорема Холла. Паросполучення у двочастковому графі

Означення 6. Паросполученням або незалежною множиною ребер у простому неорієнтованому графі $G = (V, E)$ називається така множина його ребер $M \subseteq E(G)$, елементи якого не інцидентні попарно з вершинами.

- $deg_M(v)$ — число ребер, які інцидентні вершині v , що належить множині M .
- Паросполучення — така множина ребер M , що для кожного $v \in V(G)$ справджується $deg_M(v) \leq 1$.
- Вершина називається *сполучною*, якщо вона є кінцевою точкою для ребра, що належить множині M .

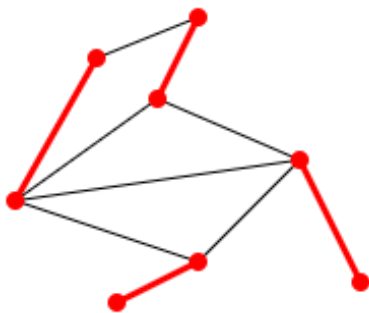


Рис. 14

Приклад графа з числом паросполучення 4 можна побачити на рис.14.

Означення 8. Паросполучення називається *максимальним*, якщо потужність M має найбільше значення з усіх можливих.

Означення 9. Паросполучення називається повним (або іноді його ще називають ідеальним), якщо кожна вершина графа є кінцем ребра, що належить паросполученню. Тоді для кожної вершини $v \in V(G)$ справджується $\deg_M(v) = 1$. Повне паросполучення є максимальним.

Висновки:

- Якщо повне паросполучення утворює граф з n вершинами, то n є парним числом.
- Якщо n є парним числом, то не обов'язково граф буде мати повне паросполучення (наприклад граф-зірка).

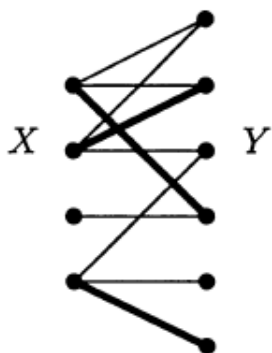


Рис. 15 [1]

Означення 10. Паросполученням у двочастковому графі $G = (V, E)$, де $V = U \cup W$ називається множина ребер, у якій ніякі два ребра не мають спільних кінцевих вершин.

Приклад двочасткового графа на рис.15, де виділені ребра позначають паросполучення.

У 1935 році Філіп Холл запропонував умову, за якої існує повне паросполучення у двочастковому графі. Представимо таку ситуацію: є n юнаків та n дівчат. Кожна дівчина має свій список вподобань, що складається з декількох юнаків, що є в початковій множині з n юнаків. Дівчата роблять пропозицію юнакам, що є у них в списку, і юнаки-джентельмени не можуть відмовитися. Така ситуація може бути представлена у вигляді двочасткового графа (рис.16), де одна частка позначає дівчат, а інша — юнаків. Ребра позначають вподобання дівчини певному юнаку.

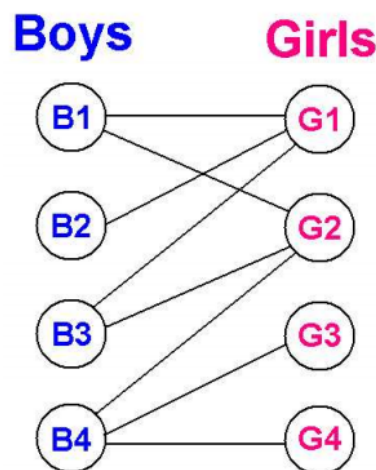


Рис. 16 [14]

Означення 11. Умова Холла про весілля справджується, коли деякій підмножині з r дівчат подобається щонайменше r юнаків.

Теорема Холла. Умова Холла про весілля є необхідною і достатньою умовою для існування повного паросполучення в двочастковому графі. Також можна стверджувати: якщо умова Холла про весілля справджується для деякого двочасткового графа, то цей двочастковий граф буде мати повне паросполучення.

Доведення:

Нехай для деякого двочасткового графа G виконується умова Холла про весілля.

Доведення проведемо за індукцією.

База: $n = 1$. Очевидно, що тоді двочастковий граф буде мати повне паросполучення. Будь-яка підмножина з одного елемента з лівої частки буде з'єднана з одним елементом з правої частки.

Крок: Припустимо, що умова Холла про весілля та теорема Холла виконується для підмножини K з лівої частки, що складається з r елементів. Доведемо, що теорема виконується для підмножини з $r + 1$ елемента. Права частка двочасткового графа буде мати підмножину $|K'| \geq r$. Розглянемо вершину з лівої частки з номером $r + 1$, яку позначимо як v_{r+1} .

Розглянемо 3 випадки:

Випадок 1. Вершина v_{r+1} з'єднана з вершиною з правої частки, що не належить підмножині K' . Тоді повне паросполучення буде існувати для $r + 1$ вершини. Приклад такого випадку можна побачити на рис.17.

Випадок 2. Вершина v_{r+1} з'єднана тільки з вершинами з підмножини K' і $|K'| = |K|$. Такий випадок суперечить умові Холла про весілля, що справджується для двочасткового графа G . Тому такий випадок є неможливим.

Випадок 3. Вершина v_{r+1} з'єднана тільки з вершинами з підмножини K'

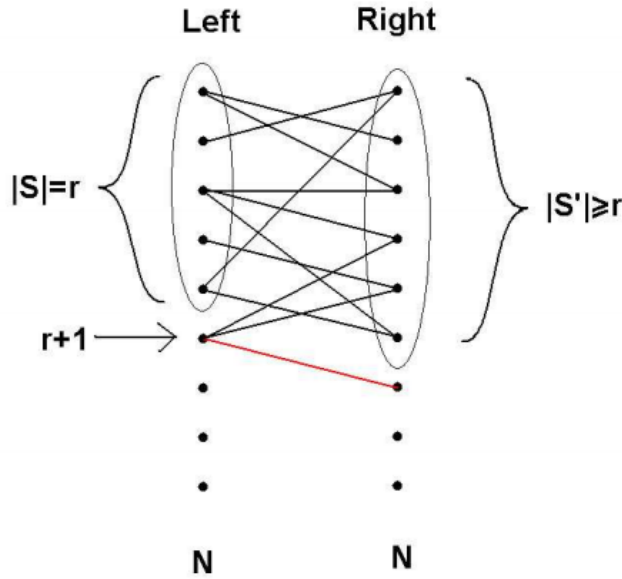


Рис. 17 [14]

і $|K'| > |K|$. Для підмножини з r елементів застосовуємо припущення про те, що для цієї підмножини буде існувати повне паросполучення. Розглянемо підмножини, що складаються з r елементів і за припущенням мають повне паросполучення: $r - 1$ елемент з підмножини K і вершина v_{r+1} . Покажемо, що тоді підмножина з $r + 1$ елементів теж буде мати повне паросполучення.

Видалимо вершину v_1 і розглянемо інші вершини з підмножини K і вершину v_{r+1} як підмножину з r елементів. Нехай ця підмножина називається K_r . Підмножина K_r утворює повне паросполучення з деякою множиною K'_r , що є підмножиною K' .

Якщо вершина v_1 з'єднана тільки з вершинами з K'_r , то це суперечить умові Холла про весілля. Отже, вершина v_1 з'єднана з деякою вершиною, що не належить K'_r , а отже підмножина з $r + 1$ вершиною буде мати повне паросполучення. Такий випадок можна побачити на рис.18.

Ребра, що виділені жирним, показують утворене повне паросполучення для вершин $v_2, \dots, v_{r+1}$. З припущення знаємо, що червоне ребро буде існувати, що доводить індукційний крок. Отже, двочастковий граф G , для якого справедлива умова Холла про весілля, буде мати повне паросполучення, а отже

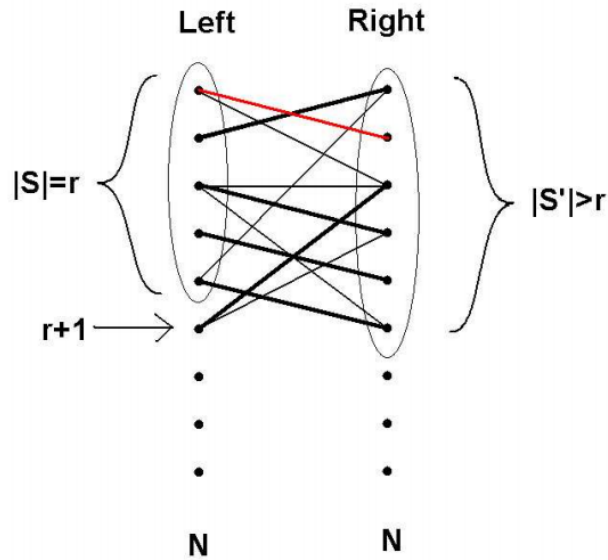


Рис. 18 [14]

для G виконується теорема Холла.

□

2.5.3 Теорема 2. Стійкі паросполучення

Проблема стійких шлюбів

Нехай задані множини: X — множина юнаків, Y — множина дівчат. Кожний з юнаків знайомий із кількома дівчатами. Ребро $uv \in E$ означає, що u і v можуть одружитися. Тоді паросполученням буде колективне весілля, на якому кожен хлопець одружений не більше одного разу. Потрібно визначити умови, за яких кожен з юнаків міг би одружитися зі знайомою йому дівчиною.

Припустимо, що в графі $G = (X \cup Y, E)$ для кожного $v \in X \cup Y$ задано деяке упорядкування множин вершин, що суміжні з вершиною v , позначимо її $N(v) = z_1 > z_2 > \dots z_{d_v}$. Тоді z_1 — вибір, що є найбажанішим для вершини v . Так само упорядкування можна зробити для кожної вершини $u \in X \cup Y$. Кожна вершина отримає своє відсортування.

Означення 12. Паросполучення M називається *стійким* в деякому графі $G = (X \cup Y, E)$, якщо для кожного ребра $uv \in E \setminus M$ такі, що $u \in X$,

$v \in Y$, виконується:

- $uy \in M$ і $y > v$ в $N(v)$
- $xv \in M$ і $x > u$ в $N(v)$
- перший, і другий пункт одночасно.

Множина весіль буде стійка, якщо в ній не буде випадків, коли $u \in N(v)$ та $v \in N(u)$ не одружені, але u надає перевагу v (одружений u або не одружений), а v надає перевагу u (заміжня v або не заміжня). Такі випадки є нестійкими, тому виключаємо їх з множини весіль, щоб знайти розв'язання задачі.

Теорема 2 (Теорема стійких шлюбів). (Д. Гейл, Л. Шеплі, 1962)

Стійке паросполучення завжди існує.

Доведення.

Розглянемо алгоритм. Нехай на першому кроці юнаки $u \in X$ роблять пропозицію найбажанішим дівчатам. Якщо дівчина отримує хоча б одну пропозицію, вона вибирає найбажанішого і вважає його майбутнім нареченим. Юнаки, що залишаються, утворюють резерв R .

Другий крок передбачає, що всі юнаки із резерву R роблять пропозицію дівчатам, що були другими у "списку бажань". На цьому кроці дівчата вже порівнюють пропозиції (і навіть порівнюють майбутніх наречених, якщо такі є), обирають найбажанішого хлопця і вважають тепер його майбутнім нареченим. Юнаки, що залишаються, утворюють резерв R .

Після, всі юнаки із резерву R роблять пропозицію дівчатам, що були третіми у "списку бажань" і так продовжується поки резерв R не стане пустим. Юнак, що зробив пропозицію останній дівчині у "списку бажань" і отримав відмову, виключається надалі зі "списку бажань" для дівчат і також виключається з резерву R . Таким чином алгоритм закінчився.

Твердження 4. Якщо алгоритм закінчився, то майбутні наречені разом з дівчатами, що їх вибрали, утворюють стійке паросполучення.

Для дівчини майбутні наречені з кожним кроком стають все більш бажанішими, так як на кожному кроці вона порівнює нові пропозиції з майбутнім нареченим і вибирає найбажанішого. Тому, якщо $uv \in E$, але $uv \notin M$, то:

- або u ніколи не робив пропозицію v (в цьому випадку u знайшов кращу наречену до того, як черга дійшла до дівчини v , отже $uy \in M$, причому $y > v$ в $N(u)$)
- або u робив пропозицію v і отримав відмову (в цьому випадку, x зробив пропозицію v раніше, за u , отже $xv \in M$, де $x > u$ в $N(v)$)

Це і є умова стійкого паросполучення. А отже, стійке паросполучення буде завжди існувати. $\square$

Алгоритм, що використовується для доведення теореми про стійке весілля і знаходження стійкого паросполучення, був запропонований Девідом Гейлом і Ллойдом Шеплі у **1962** році. Лише через 30 років Елвін Рот почав застосовувати цей алгоритм в соціальних проектах. На сьогодні цей алгоритм залишається актуальним у США та ще деяких країнах. Наприклад, розподіл старшокурсників по школах, лікарів-початківців в лікарні і також підбір донорів для пересадки нирки, - мають в основі алгоритм Гейла-Шеплі. У 2012 році Шеплі і Роту (Гейл на той час вже помер) була присуджена Нобелівська премія за створення і впровадження в практику теорії оптимального розподілу.

Приклад утворення максимального паросполучення в двочастковому графі, яке не є стійким:

На рис.19 виділені жирним ребра утворюють стійке паросполучення. Кожний рядок показує пріоритетність вибору.

Розглянемо детальніше всі паросполучення. Нехай a, b, c, d — множина юнаків, а $\{A, B, C, D\}$ — множина дівчат. Застосуємо алгоритм Гейла-Шеплі, щоб знайти стійке паросполучення:

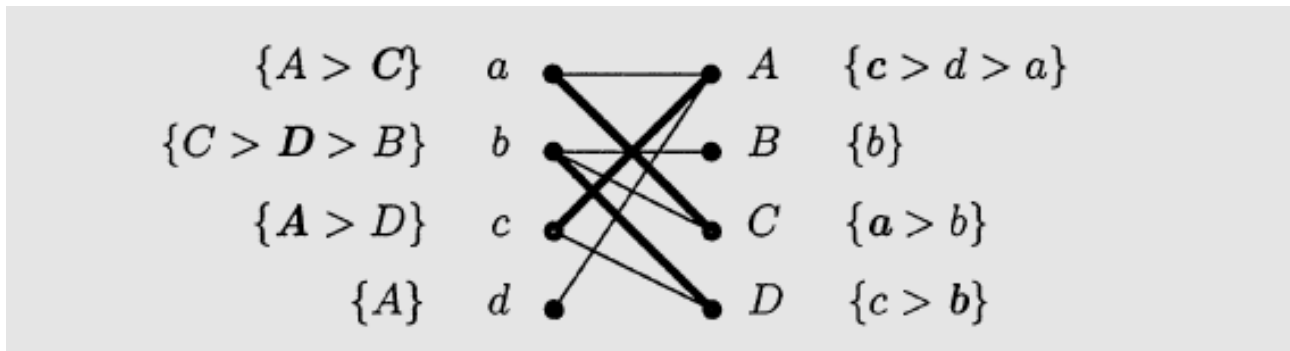


Рис. 19 [1]

Крок 1. Юнаки роблять пропозицію найпріоритетнішим дівчатам у своєму "списку бажань". Тобто a робить пропозицію A , b — C , c — A , d — A . C погоджується на пропозицію b , а A обирає між трьома юнаками, обирає найкращого вже зі свого "списку бажань". Таким чином, утворюються пари: $\{bC, cA\}$. Резерв з юнаків: $\{a\}$. Юнак d не входить до резерву і не має пари, бо у нього в "списку бажань" була тільки дівчина A , що погодилася на пропозицію більш пріоритетнішого.

Крок 2. Юнак з резерву a робить пропозицію наступній дівчині у своєму "списку бажань". Дівчина C вже прийняла пропозицію від b , але a стоїть пріоритетніше. Таким чином, утворюються пари: $\{aC, cA\}$. Резерв з юнаків: $\{b\}$.

Крок 3. Юнак з резерву a робить пропозицію наступній дівчині у своєму "списку бажань". Дівчина D погоджується, так як їй ще ніхто не робив пропозицію. Таким чином, утворюються пари: $\{aC, bD, cA\}$. Резерв з юнаків стає пустим.

Отже, $M = \{aC, bD, cA\}$ і це паросполучення стійке.

Знайдемо максимальне паросполучення. Паросполучення у двочастковому графі буде максимальним, якщо жодні два ребра не мають спільну вершину. Максимальне паросполучення повинно мати максимальну кількість ребер

зі всіх можливих паросполучень.

Так як юнак d знайомий тільки з дівчиною A , то утворюємо пару dA . Якщо d не утворить пару, то паросполучення вже не буде максимальним, так як буде мати не максимальну кількість ребер.

Дівчина B знайома тільки з юнаком b , а отже утворюємо пару bB . Якщо B не утворить пару, то паросполучення вже не буде максимальним, так як буде мати не максимальну кількість ребер. Таким чином маємо вже дві утворені пари: $\{bB, dA\}$.

Обидва юнаки a і c знайомі з дівчиною A , але дівчина A вже утворює пару. Юнак a знайомий крім A тільки з дівчиною C , а отже утворюється пара aC . Юнак c знайомий крім A тільки з дівчиною D , а отже утворюється пара cD .

Цей приклад показує, що маємо єдине максимальне паросполучення $M = \{aC, bB, cD, dA\}$ з чотирма ребрами, але воно нестійке через ребро cA .

Стійке паросполучення буде існувати для заданого двочасткового графа, але воно буде не єдиним. Алгоритм Гейла-Шеплі повертає стійке паросполучення, яке є оптимальним для юнаків. Також алгоритм може повернути стійке паросполучення, яке буде оптимальним для дівчат. В такому випадку пропозиції повинні робити дівчата. Ці два стійких паросполучення будуть верхніми і нижніми елементами **решітки стійких паросполучень**.

Означення 13. *Решітка стійких паросполучень* — дистрибутивна решітка, елементами якої є стійкі паросполучення.

Приклад знайомств, що будуть утворювати решітку стійких паросполучень:

"Списки бажань" для кожної з підмножин юнаків та дівчат, можна побачити на рис.20.

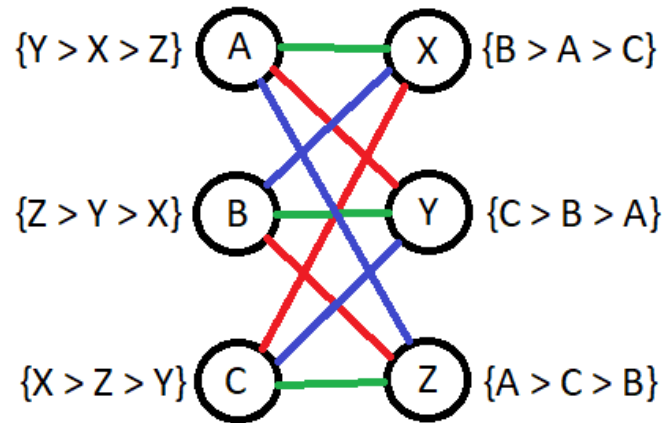


Рис. 20

Верхній елемент решітки — оптимальне стійке паросполучення для юнаків. Юнаки роблять пропозицію найбажанішим дівчатам у своєму "списку бажань". А робить пропозицію Y, B — Z, C — X. Таким чином утворюються пари $\{AY, BZ, CX\}$.

Отже, $M_1 = \{AY, BZ, CX\}$ — це максимальне паросполучення і воно є оптимально стійким для юнаків, так як немає таких паросполучень, що одружені, але обидва надають перевагу іншим.

Нижній елемент решітки — оптимальне стійке паросполучення для дівчат. Дівчата роблять пропозицію найбажанішим юнакам у своєму "списку бажань". X робить пропозицію B, Y — C, A — Z. Таким чином утворюються пари $\{XB, YC, AZ\}$.

Отже, $M_2 = \{XB, YC, AZ\}$ — це максимальне паросполучення і воно є оптимально стійким для дівчат, так як немає таких паросполучень, що одружені, але обидва надають перевагу іншим.

Середній елемент решітки — оптимально стіке для юнаків і дівчат одночасно. Дівчата і юнаки в такому випадку обирають другого за пріорите-

том у своєму "списку бажань". Таким чином утворюються пари $\{AX, BY, CZ\}$.

Отже, $M_3 = \{AX, BY, CZ\}$ — це *максимальне паросполучення і воно є оптимально стійким для юнаків і дівчат*, так як немає таких паросполучень, що одружені, але обидва надають перевагу іншим.

Всі ці три стійких паросполучення $\{M_1, M_2, M_3\}$ будуть утворювати решітку стійких паросполучень для заданого прикладу знайомств.

2.6 $\chi_l(S_n) = n$

Щоб отримати розв'язання задачі Діниця, яке знайшов Галвін, треба застосувати разом теореми 1 і 2.

Теорема 3 (Теорема Діниця). Для всіх n буде справедлива рівність $\chi_l(S_n) = n$.

Доведення.

Нехай вершини позначимо, як (i, j) , де $1 \leq i, j \leq n$. Вершини (i, j) та (r, s) будуть суміжними тоді і тільки тоді, коли $i = r$ або $j = s$.

Розглянемо довільний латинський квадрат з числами $\{1, 2, \dots, n\}$ і позначимо $L(i, j)$ — число в клітині (i, j) . Далі потрібно зробити перетворення графа S_n в орієнтований граф $\vec{S}_n$. Горизонтальні ребра орієнтуємо так:

$$(i, j) \longrightarrow (i, j'), \text{ якщо } L(i, j) < L(i, j')$$

Вертикальні ребра орієнтуємо так:

$$(i, j) \longrightarrow (i', j), \text{ якщо } L(i, j) > L(i', j)$$

На рис.21 можна побачити приклад орієнтованого графа $\vec{S}_3$. За латинським квадратом отримуємо орієнтацію. Горизонтальні ребра орієнтуються від меншого числа до більшого, вертикальні ребра навпаки від більшого числа до меншого.

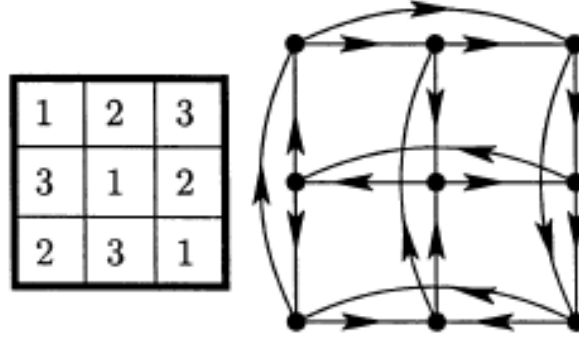


Рис. 21 [1]

Степінь виходу для кожної вершини графа $\vec{S}_n$ буде дорівнювати $n - 1$, так як якщо вершина суміжна з $n - 1$ вершиною по горизонталі і буде мати $n - 1$ ребро виходу, то ця вершина буде з найменшим числом в латинському квадраті, що відповідає цьому графу, а отже по вертикалі вершина не буде мати ребер виходу. Звідки $d_{(i,j)}^+ = n - 1$ для кожної клітини (i, j) .

Треба показати, що кожний індукований підграф орграфа $\vec{S}_n$ має ядро, застосовуючи теорему 1.

Розглянемо множину клітин A . Нехай X — множина рядків латинського квадрата L , Y — множина стовпців. Зіставимо множині клітин A двочастковий граф $G = (X \cup Y, A)$, кожна пара $(i, j) \in A$ відповідає ребру графа ij , де $i \in X$, $j \in Y$. На рис.22 можна побачити, як з латинського квадрата розмірності 4×4 вибрали множину клітин A (клітини позначені сірим кольором) і зіставили цій множині двочастковий граф.

Перетворення графа S_n в орієнтований граф $\vec{S}_n$ породжує порядок в $N(i)$ вершин графа $G = (X \cup Y, A)$ за таким принципом:

- $j' > j$ в $N(i)$, $i \in X$, якщо $(i, j) \longrightarrow (i, j')$ в $\vec{S}_n$
- $i' > i$ в $N(j)$, $j \in Y$, якщо $(i, j) \longrightarrow (i', j)$ в $\vec{S}_n$

За теоремою 2 граф $G = (X \cup Y, A)$ має стійке паросполучення M . Множина M — це шукане ядро орграфа $\vec{S}_n$, що індукований множиною A . Це

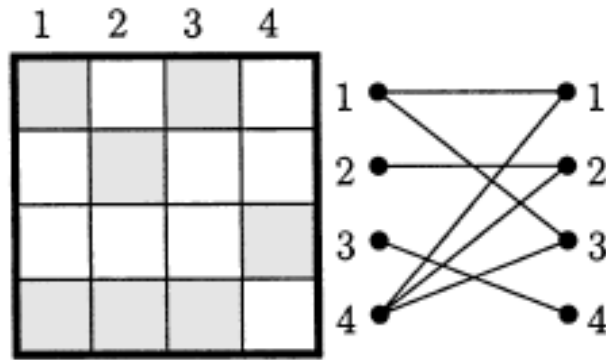


Рис. 22 [1]

дійсно виконується, бо множина M є незалежною в орграфі $\vec{S}_n$, оскільки елементи M , що розглядаються як ребра в графі $G = (X \cup Y, A)$, не мають спільних кінців i або j .

Також, якщо $(i, j) \in A \setminus M$, то або $(i, j') \in M$ з $j' > j$, або $(i', j) \in M$ з $i' > i$. Для орграфа $\vec{S}_n$ це означає, що або дуга $(i, j) \rightarrow (i', j) \in M$, або дуга $(i, j) \rightarrow (i, j') \in M$. Доведення завершено. $\square$

2.7 Практична частина

2.7.1 Алгоритм знаходження розфарбування для задачі Діниці

На основі доведення теорем 1, 2 та 3 формуємо такий алгоритм знаходження розфарбування за списком квадрата, що використовується в задачі Діниці. В загальному випадку розглядаємо квадрат розміру $n \times n$.

1	2	3	4	...	n
n	1	2	3	4	...
n-1	n	1	2	3	...
n-2	n-1	n	1	2	...
...	n-2	n-1	n	1	2
2	...	n-2	n-1	n	1

Рис. 23

Крок 1: Будуємо латинський квадрат $n \times n$, що будемо заповнювати елементами з множини натуральних чисел $\{1, 2, \dots, n\}$. Наприклад, використаємо діагональний метод заповнення для утворення повного латинського квадрата $n \times n$. Такий квадрат можна побачити на рис.23.

Крок 2: Будуємо граф S_n на основі утвореного латинського квадрата. Нехай граф S_n складається з множини вершин V . Елементи латинського квадрата будуть вершинами, тому потужність множини $|V| = n^2$ і $V = \{1, 2, \dots, n^2\}$. Отриманий граф S_n можна побачити на рис.24.

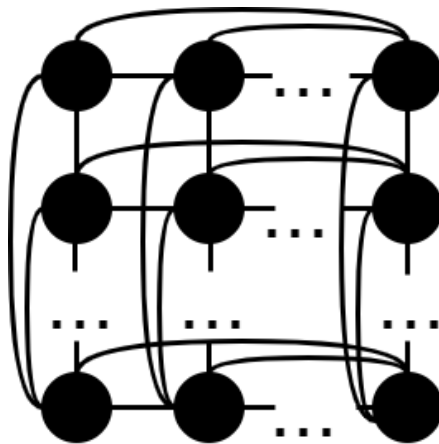


Рис. 24

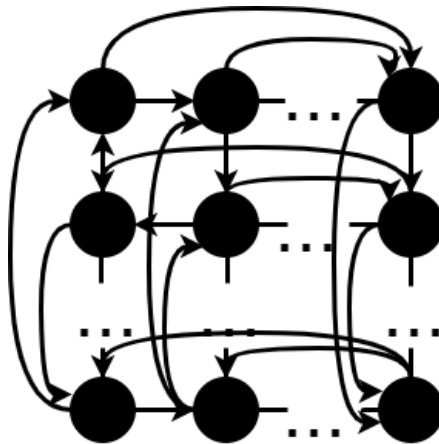


Рис. 25

Крок 3: Перетворюємо граф S_n в оргграф $\vec{S}_n$. За теоремою 3 орієнтація робиться так: горизонтальні ребра від найменшого числа до найбільшого, вертикальні ребра — від найбільшого до найменшого. Отриманий оргграф $\vec{S}_n$

можна побачити на рис.25.

Крок 4: Робимо розфарбування за теоремою 1. Для кожної вершини орграфа $\vec{S}_n$ $v \in V$ задана множина кольорів $C(v)$. $C = \bigcup_{v \in V} C(v)$. Потужність множини кольорів $|C| = m$, $m > n$. На множині C обираємо довільну нумерацію.

1. Вибираємо колір C_1 з множини C .
2. Утворюємо множину $A(C_1)$ — множина, що складається з усіх вершин графа $\vec{S}_n$, що мають у своєму списку кольорів для розфарбування колір C_1 . Тобто $A(C_1) := \{v \in V : C_1 \in C(v)\}$.
3. В графі, індукованому множиною $A(C_1)$, який назвемо G_1 , знайдемо ядро $K(C_1)$.
4. Фарбуємо всі вершини ядра в колір C_1 . Ядро — незалежна множина, тому розфарбування цих вершин не суперечить умові задачі Діниця.
5. Видаляємо ядро $K(C_1)$ з $\vec{S}_n$.
6. Вибираємо колір C_2 з множини $C \setminus C_1$.
7. Утворюємо множину $A(C_2)$ — множина, що складається з усіх вершин графа $\vec{S}_n \setminus K(C_1)$, що мають у своєму списку кольорів для розфарбування колір C_2 . Тобто $A(C_2) := \{v \in V \setminus K(C_1) : C_2 \in C(v)\}$.
8. В графі, індукованому множиною $A(C_2)$, який назвемо G_2 , знайдемо ядро $K(C_2)$.
9. Фарбуємо всі вершини ядра в колір C_2 .
10. Видаляємо ядро $K(C_2)$ з $\vec{S}_n$.
- ...
11. Вибираємо колір C_k з множини $C \setminus \bigcup_{i=1}^{k-1} C_i$.

12. Утворюємо множину $A(C_k)$ — множина, що складається у всіх вершин графа $\vec{S}_n \setminus \bigcup_{i=1}^{k-1} K(C_i)$, що мають у своєму списку кольорів для розфарбування колір C_k . Тобто $A(C_k) := \{v \in V \setminus \bigcup_{i=1}^{k-1} K(C_i) : C_k \in C(v)\}$.
13. В графі, індукованому множиною $A(C_k)$, який назовемо G_k , знайдемо ядро $K(C_k)$.
14. Фарбуємо всі вершини ядра в колір C_k .
15. Видаляємо ядро $K(C_k)$ з $\vec{S}_n$.

Повторюємо такі кроки, поки граф $\vec{S}_n$ не буде повністю розфарбованим.

Таким чином отримуємо правильне розфарбування за списком графа S_n , яке і є розв'язанням задачі Діниця.

2.7.2 Задача 1.1

Застосуємо алгоритм до графа S_3 :

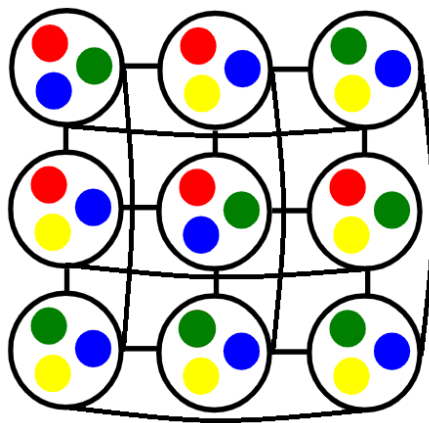


Рис. 26

Нехай граф S_3 буде мати множину кольорів C , що складається з чотирьох кольорів, тобто $C = \bigcup_{v \in V} C(v) = 4$. Множини кольорів для кожної клітини

можна побачити на рис.26. За теоремою Діниця знаємо, що $\chi_l(S_3) = 3$, тому $|C(v)| = 3 \ \forall v \in V$.

Розглянемо деякий латинський квадрат 3×3 . Перетворимо граф S_3 в $\vec{S}_3$, орієнтуємо горизонтальні ребра від найменшого числа до найбільшого, вертикальні ребра — від найбільшого до найменшого (рис.27). Потужність множини кольорів більша степені виходу для кожної вершини, тобто $|C(v)| \geq d_v^+ + 1 \ \forall v \in V$.

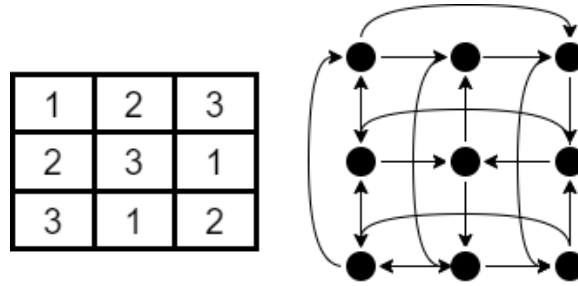


Рис. 27

Крок 1: Розглянемо синій колір і всі вершини з орієнтовними ребрами, що мають в своїй множині кольорів синій. Отриманий підграф можна побачити на рис.28. Застосуємо означення ядра, звідки отримуємо, що ядром цього підграфа будуть вершини **(1,3)**, **(2,2)**, **(3,1)**. На рис.28 вершини з ядра виділені жирним. В початковому графі S_3 фарбуємо вершини **(1,3)**, **(2,2)**, **(3,1)** в синій колір. В орграфі $\vec{S}_3$ видаляємо ці вершини разом з ребрами, що виходять з цих вершин. І далі вже розглядаємо $\vec{S}_3$ без цих вершин.

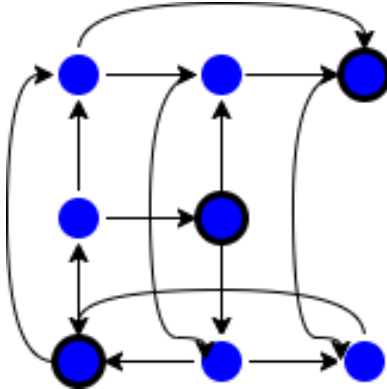


Рис. 28

Крок 2: Розглянемо жовтий колір і всі вершини з орієнтовними ребрами,

що мають в своїй множині кольорів жовтий. Отриманий підграф можна побачити на рис.29. Застосуємо означення ядра, звідки отримуємо, що ядром цього підграфа будуть вершини **(1,2)**, **(2,1)**, **(3,3)**. На рис.29 вершини з ядра виділені жирним. В початковому графі S_3 фарбуємо вершини **(1,2)**, **(2,1)**, **(3,3)** в жовтий колір. В оргграфі $\vec{S}_3$ видаляємо ці вершини разом з ребрами, що виходять з цих вершин. І далі вже розглядаємо $\vec{S}_3$ без цих вершин.

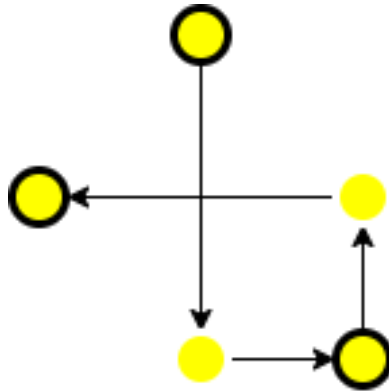


Рис. 29

Крок 3: Розглянемо червоний колір і всі вершини, що мають в своїй множині кольорів червоний. Отриманий підграф можна побачити на рис.30. Застосуємо означення ядра, звідки отримуємо, що ядром цього підграфа будуть обидві вершини цього підграфа **(1,1)**, **(2,3)**. На рис.30 вершини з ядра виділені жирним. В початковому графі S_3 фарбуємо вершини **(1,1)**, **(2,3)** в червоний колір. В оргграфі $\vec{S}_3$ видаляємо ці вершини разом з ребрами, що виходять з цих вершин. І далі вже розглядаємо $\vec{S}_3$ без цих вершин.



Рис. 30

Крок 4: Розглянемо зелений колір і останню непофарбовану вершину графа S_3 , що має в своїй множині кольорів зелений. Фарбуємо вершину **(3,2)** в зелений. На цьому закінчуємо розфарбування і отримуємо розфарбування за списком для графа S_3 , що можна побачити на рис.31.

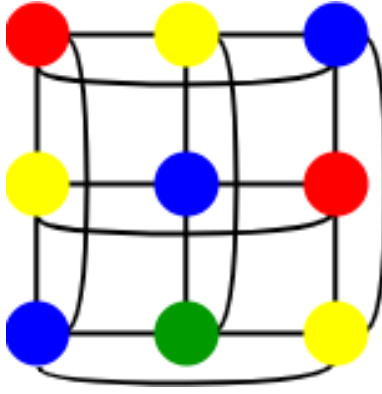


Рис. 31

2.7.3 Задача 1.2

Застосуємо алгоритм до графа S_5 :

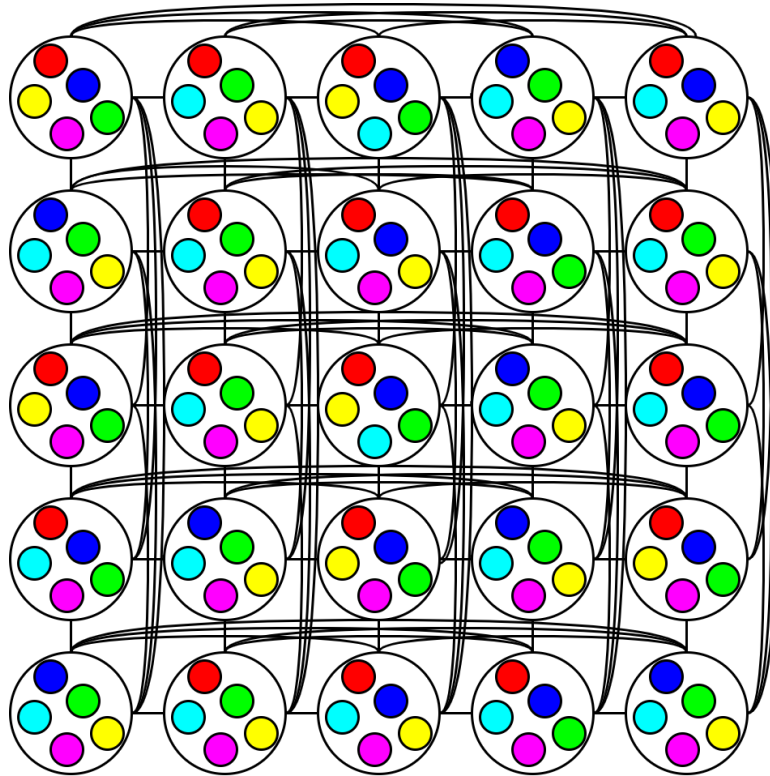


Рис. 32

Нехай граф S_5 буде мати множину кольорів C , що складається з шести кольорів, тобто $C = \bigcup_{v \in V} C(v) = 6$. Множини кольорів для кожної клітини можна побачити на рис.32. За теоремою Діниця знаємо, що $\chi_l(S_5) = 5$, тому $|C(v)| = 3 \forall v \in V$.

Розглянемо деякий латинський квадрат 5×5 , використовуємо діагональний метод заповнення. Перетворимо граф S_5 в $\vec{S}_5$, орієнтуємо горизонтальні ребра від найменшого числа до найбільшого, вертикальні ребра — від найбільшого до найменшого (рис.33). Потужність множини кольорів більша степені виходу для кожної вершини, тобто $|C(v)| \geq d_v^+ + 1 \forall v \in V$.

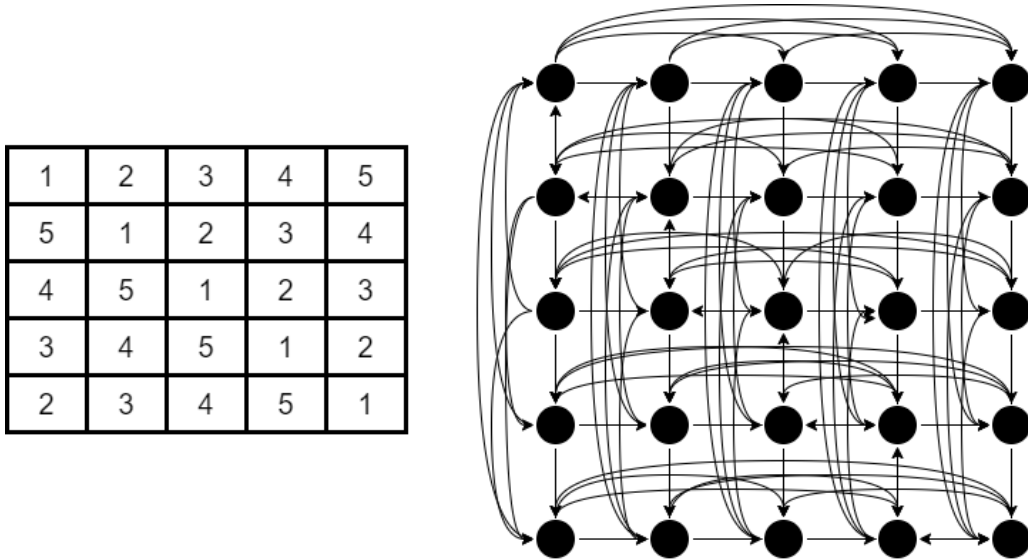


Рис. 33

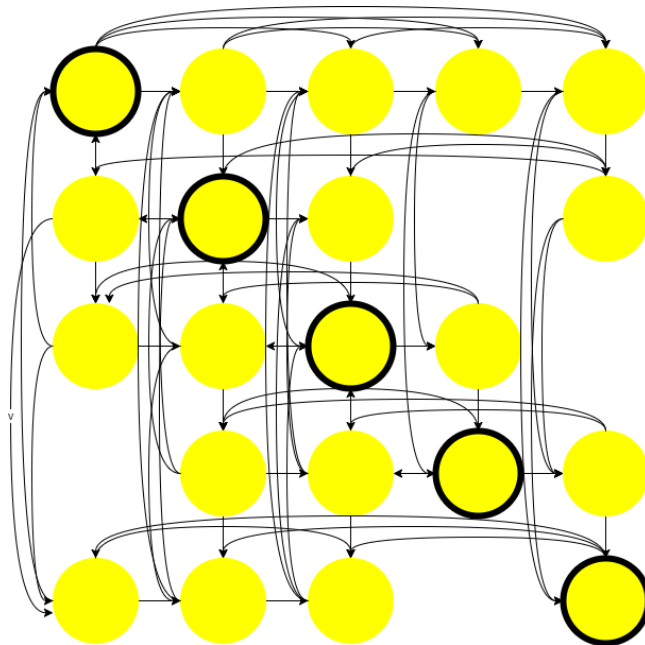


Рис. 34

Крок 1: Розглянемо жовтий колір і всі вершини з орієнтовними ребрами, що мають в своїй множині кольорів жовтий. Отриманий підграф можна по-

бачити на рис.34. Застосуємо означення ядра, звідки отримуємо, що ядром цього підграфа будуть вершини $(1,1)$, $(2,2)$, $(3,3)$, $(4,4)$, $(5,5)$. На рис.34 вершини з ядра виділені жирним. В початковому графі S_5 фарбуємо вершини $(1,1)$, $(2,2)$, $(3,3)$, $(4,4)$, $(5,5)$ в жовтий колір. В оргграфі $\vec{S}_5$ видаляємо ці вершини разом з ребрами, що виходять з цих вершин. І далі вже розглядаємо $\vec{S}_5$ без цих вершин.

Крок 2: Розглянемо червоний колір і всі вершини з орієнтовними ребрами, що мають в своїй множині кольорів червоний. Отриманий підграф можна побачити на рис.35. Застосуємо означення ядра, звідки отримуємо, що ядром цього підграфа будуть вершини $(1,3)$, $(2,5)$, $(3,2)$, $(4,1)$, $(5,4)$. На рис.35 вершини з ядра виділені жирним. В початковому графі S_5 фарбуємо вершини $(1,3)$, $(2,5)$, $(3,2)$, $(4,1)$, $(5,4)$ в червоний колір. В оргграфі $\vec{S}_5$ видаляємо ці вершини разом з ребрами, що виходять з цих вершин. І далі вже розглядаємо $\vec{S}_5$ без цих вершин.

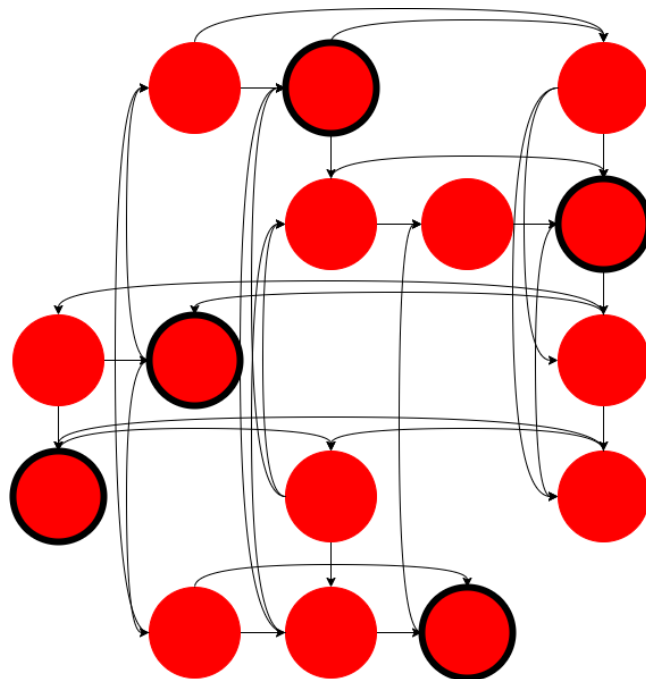


Рис. 35

Крок 3: Розглянемо синій колір і всі вершини з орієнтовними ребрами, що мають в своїй множині кольорів синій. Отриманий підграф можна побачити на рис.36. Застосуємо означення ядра, звідки отримуємо, що ядром цього підграфа будуть вершини $(1,5)$, $(2,4)$, $(3,1)$, $(4,2)$, $(5,3)$. На рис.36 вершини

з ядра виділені жирним. В початковому графі S_5 фарбуємо вершини **(1,5)**, **(2,4)**, **(3,1)**, **(4,2)**, **(5,3)** в синій колір. В орграфі $\vec{S}_5$ видаляємо ці вершини разом з ребрами, що виходять з цих вершин. І далі вже розглядаємо $\vec{S}_5$ без цих вершин.

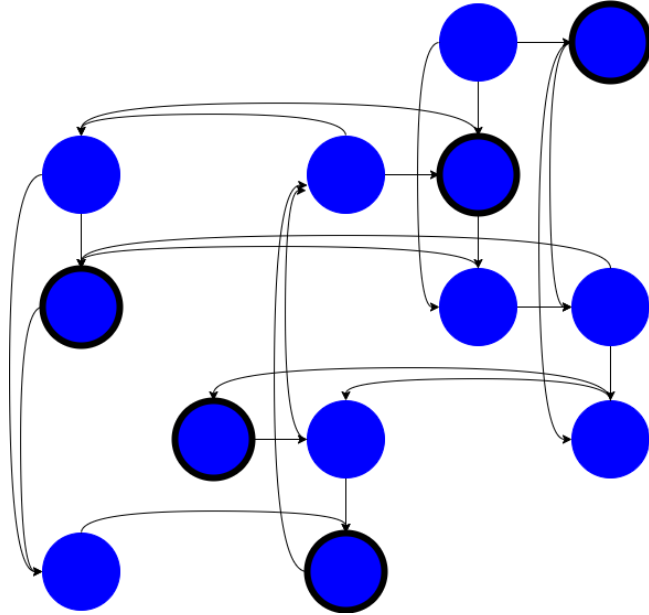


Рис. 36

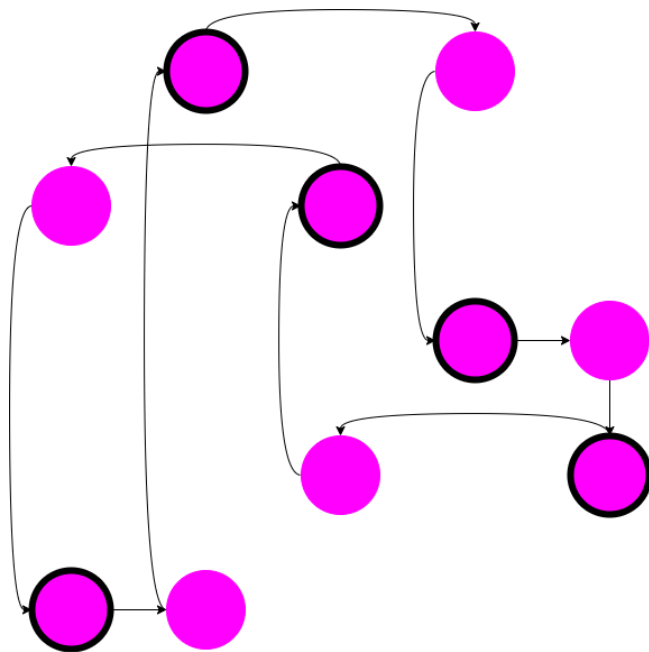


Рис. 37

Крок 4: Розглянемо рожевий колір і всі вершини з орієнтовними ребрами, що мають в своїй множині кольорів рожевий. Отриманий підграф можна побачити на рис.36. Застосуємо означення ядра, звідки отримуємо, що ядром

цього підграфа будуть вершини $(1,2)$, $(2,3)$, $(3,4)$, $(4,5)$, $(5,1)$. На рис.36 вершини з ядра виділені жирним. В початковому графі S_5 фарбуємо вершини $(1,2)$, $(2,3)$, $(3,4)$, $(4,5)$, $(5,1)$ в рожевий колір. В оргграфі $\vec{S}_5$ видаляємо ці вершини разом з ребрами, що виходять з цих вершин. І далі вже розглядаємо $\vec{S}_5$ без цих вершин.

Крок 5: Розглянемо блакитний колір і всі вершини з орієнтовними ребрами, що мають в своїй множині кольорів блакитний. Отриманий підграф можна побачити на рис.36. Застосуємо означення ядра, звідки отримуємо, що ядром цього підграфа будуть вершини $(1,4)$, $(2,1)$, $(3,5)$, $(5,2)$. На рис.36 вершини з ядра виділені жирним. В початковому графі S_5 фарбуємо вершини $(1,4)$, $(2,1)$, $(3,5)$, $(5,2)$ в блакитний колір. В оргграфі $\vec{S}_5$ видаляємо ці вершини разом з ребрами, що виходять з цих вершин. І далі вже розглядаємо $\vec{S}_5$ без цих вершин.

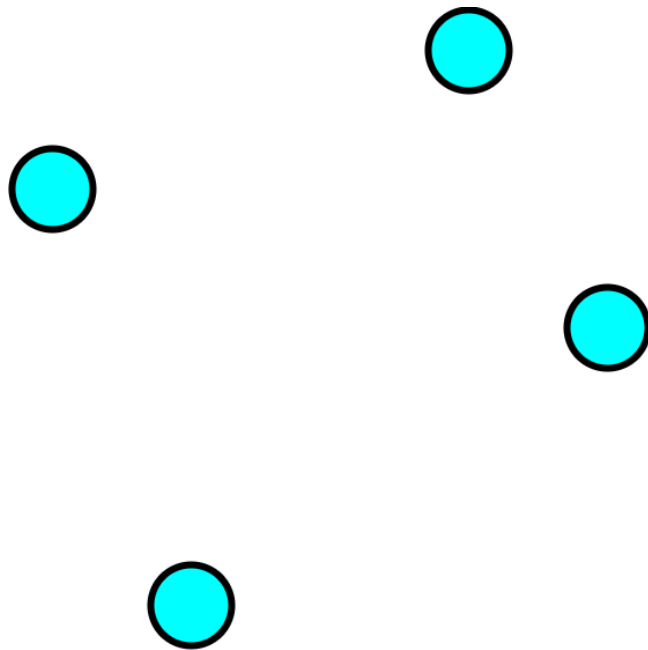


Рис. 38

Крок 6: Розглянемо зелений колір і останню непофарбовану вершину графа S_5 , що має в своїй множині кольорів зелений. Фарбуємо вершину $(4,3)$ в зелений. На цьому закінчуємо розфарбування і отримуємо розфарбування за списком для графа S_5 , що можна побачити на рис.39.

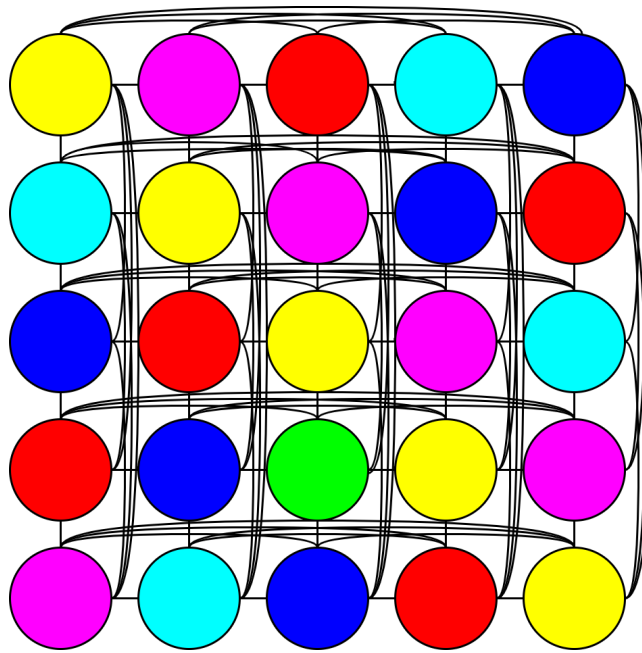


Рис. 39

2.7.4 Задача 2.1

Застосуємо теорему 2 та алгоритм Гейла-Шеплі. Покажемо, що стійке паросполучення буде існувати для деякого індукованого підграфа орграфа $\vec{S}_3$:

Розглянемо деяку множину клітин R орграфа $\vec{S}_3$, яка виділена сірим на рис.40. Зіставляємо множині R двочастковий граф.

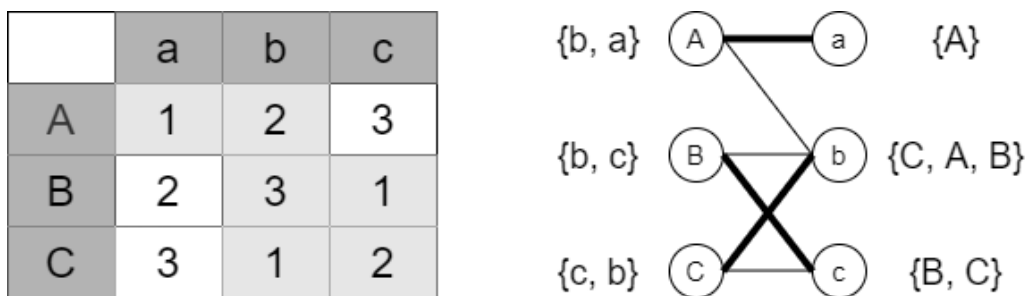


Рис. 40

Нехай множина $\{A, B, C\}$ — множина юнаків, а $\{a, b, c\}$ — множина дівчат. Тоді знаходження стійкого паросполучення можна переформулювати у "проблему стійких шлюбів". Сформульований "список бажань" для юнаків і дівчат можна побачити на рис.40. Застосуємо алгоритм Гейла-Шеплі і покажемо, що отриманий двочастковий граф, що складається з усіх можливих вподобань юнаків і дівчат, буде мати стійке паросполучення:

Крок 1: Юнаки роблять пропозицію дівчатам, що стоять першими у "списку бажань". Дівчата погоджуються на кращого в своєму "списку бажань". Так, утворюються пари: $\{Ab, Cc\}$. Резерв з юнаків, що не отримали пару: $\{B\}$.

Крок 2: Юнак B робить пропозицію другій дівчині у своєму списку. Дівчина c вже має пару з юнаком C , але для неї кращим є B . Так, утворюються пари: $\{Ab, Bc\}$. Резерв з юнаків, що не отримали пару: $\{C\}$.

Крок 3: Юнак C робить пропозицію другій дівчині у своєму списку. Дівчина b вже має пару з юнаком A , але для неї кращим є C . Так, утворюються пари: $\{Bc, Cb\}$. Резерв з юнаків, що не отримали пару: $\{A\}$.

Крок 4: Юнак A робить пропозицію другій дівчині у своєму списку. Дівчина a погоджується. Так, утворюються пари: $\{Aa, Bc, Cb\}$. Резерв з юнаків, що не отримали пару, стає пустим, тому алгоритм закінчується.

На рис.40 ребра стійкого паросполучення виділені жирним. Отримане стійке паросполучення — це і є ядро орграфа $\vec{S}_3$, що індукований множиною R . Дійсно, для графа S_3 будуть виконуватися теореми 1 і 2.

2.7.5 Задача 2.2

Застосуємо теорему 2 та алгоритм Гейла-Шеплі. Покажемо, що стійке паросполучення буде існувати для деякого індукованого підграфа орграфа $\vec{S}_5$:

Розглянемо деяку множину клітин R орграфа $\vec{S}_5$, яка виділена сірим на рис.41. Зіставляємо множині R двочастковий граф.

Нехай множина $\{A, B, C, D, E\}$ — множина юнаків, а $\{a, b, c, d, e\}$ — множина дівчат. Тоді знаходження стійкого паросполучення можна переформулювати у "проблему стійких шлюбів". Сформульований "список бажань" для

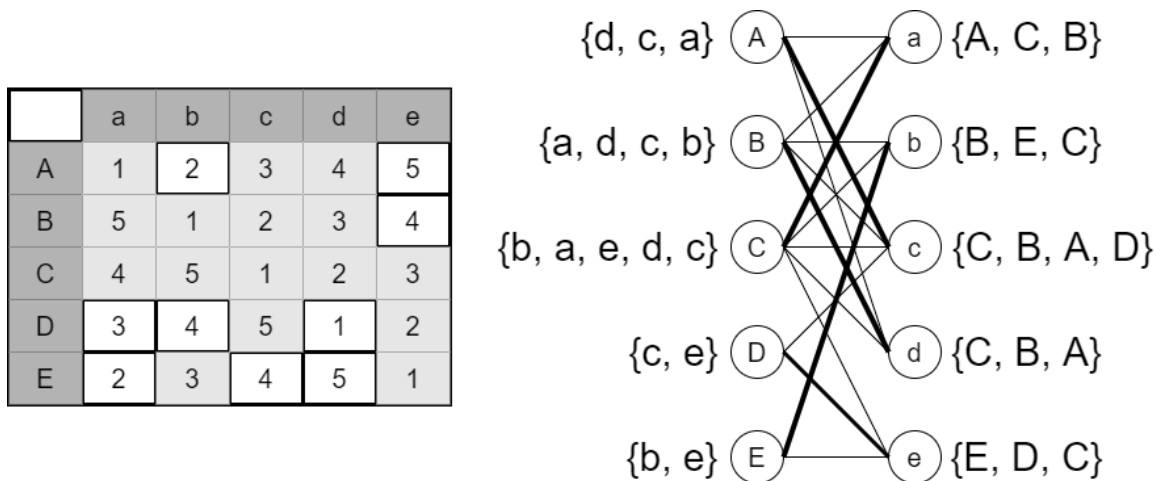


Рис. 41

юнаків і дівчат можна побачити на рис.41. Застосуємо алгоритм Гейла-Шеплі і покажемо, що отриманий двочастковий граф, що складається з усіх можливих вподобань юнаків і дівчат, буде мати стійке паросполучення:

Крок 1: Юнаки роблять пропозицію дівчатам, що стоять першими у "списку бажань". Дівчата погоджуються на кращого в своєму "списку бажань". Так, утворюються пари: $\{Ad, Ba, Dc, Eb\}$. Резерв з юнаків, що не отримали пару: $\{C\}$.

Крок 2: Юнак C робить пропозицію другій дівчині у своєму списку. Дівчина a вже має пару з юнаком B , але для неї кращим є C . Так, утворюються пари: $\{Ad, Ca, Dc, Eb\}$. Резерв з юнаків, що не отримали пару: $\{B\}$.

Крок 3: Юнак B робить пропозицію другій дівчині у своєму списку. Дівчина d вже має пару з юнаком A , але для неї кращим є B . Так, утворюються пари: $\{Bd, Ca, Dc, Eb\}$. Резерв з юнаків, що не отримали пару: $\{A\}$.

Крок 4: Юнак A робить пропозицію другій дівчині у своєму списку. Дівчина c вже має пару з юнаком D , але для неї кращим є A . Так, утворюються пари: $\{Ac, Bd, Ca, Eb\}$. Резерв з юнаків, що не отримали пару: $\{D\}$.

Крок 5: Юнак D робить пропозицію другій дівчині у своєму списку. Дівчина e погоджується. Так, утворюються пари: $\{Ac, Bd, Ca, De, Eb\}$. Резерв з

юнаків, що не отримали пару, стає пустим, тому алгоритм закінчується.

На рис.41 ребра стійкого паросполучення виділені жирним. Отримане стійке паросполучення — це і є ядро орграфа $\vec{S}_5$, що індукований множиною R . Дійсно, для графа S_5 будуть виконуватися теореми 1 і 2.

2.8 Гіпотеза про хроматичні числа реберних графів

Означення 14. $K_{n,n}$ — повний двочастковий граф, який складається з множини вершин $X \cup Y$, $|X| = |Y| = n$ і множини ребер і множиною ребер $\{(u, v) : u \in X, v \in Y\}$.

Означення 15. Для будь-якого графа G можна визначити *реберний граф* G_E . Множина вершин цього графа — множина ребер графа G , тобто $V(G_E) = E(G)$. Дві вершини $u, v \in V(G_E)$ суміжні тоді і тільки тоді, коли відповідні ребра графа G мають спільну вершину.

Твердження 5. S_n — реберний граф графа $K_{n,n}$.

Доведення.

З означення графа S_n знаємо, що це таблиця $n \times n$. Граф $K_{n,n}$ складається з двох часток з кількістю вершин n , тому можемо представити цей граф, як: ліва частка — це рядки таблиці, а права частка — стовпці. Таким чином, граф $K_{n,n}$ пов'язується з таблицею $n \times n$, а отже і з графом S_n .

Нехай деяка вершина u з лівої частки графа $K_{n,n}$ з'єднується ребром з деякою вершиною v з правої частки. В таблиці $n \times n$ u -тому елементу в рядку та v -тому елементу в стовпці відповідає вершина (u, v) , що стоїть на перетині рядка і стовпця. Ребра, що з'єднують вершини в графі S_n відповідають вершинам з графа $K_{n,n}$. Якщо в графі $K_{n,n}$ два ребра мають спільну вершину, то в графі S_n відповідні дві вершини з'єднуються ребром. За означенням реберного графа, S_n буде реберним графом графа $K_{n,n}$. $\square$

Приклад реберного графа $K_{4,4}$:

На рис.42 справа можна побачити граф $K_{4,4}$, зліва — відповідний йому реберний граф S_4 .

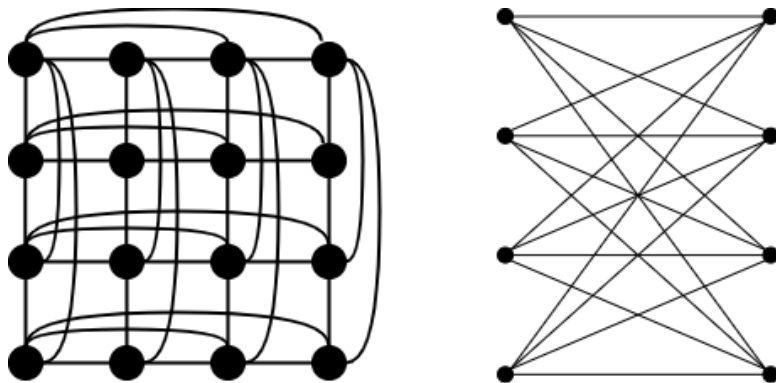


Рис. 42

В загальному випадку позначатимемо реберний граф H і тоді $H = L(G)$ для деякого графа G . Не кожний граф буде реберним для якогось графа.

Приклад графа $K_{2,4}$, який не буде реберним ні для якого графа:

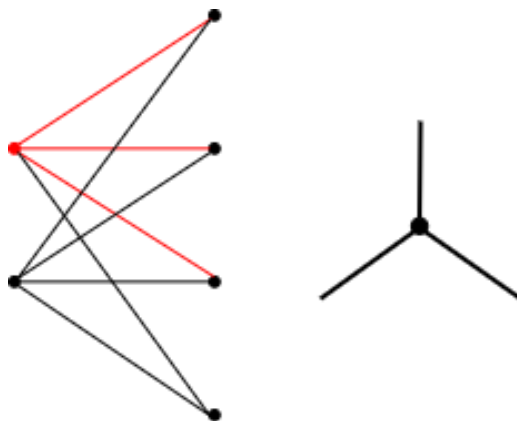


Рис. 43

На рис.43 зліва зображений граф $K_{2,4}$, справа — мінімальний нереберний граф. Характеристика реберного графа, що була доведена Байнеке: граф буде реберним тоді і тільки тоді, коли жодна з підмножин цього графа не є одним з дев'яти мінімальних нереберних графів. Граф $K_{2,4}$ має підмножину, що є мінімальним нереберним графом і виділена червоним на рис.43, тому граф $K_{2,4}$ не є реберним.

Якщо H — реберний граф двочасткового графа, то доведеним є твердження про те, що $\chi(H) = \chi_l(H)$. Головна гіпотеза досі залишається не розв’язаною в загальному випадку: *чи правильним є твердження, що $\chi(H) = \chi_l(H)$ для будь-якого реберного графа H ?*

2.9 Прикладне застосування

Розфарбування графів має широке прикладне застосування, починаючи від складання розкладів до розподілу регістрів в мікропроцесах. Розфарбування за списком графів застосовується у всіх задачах, що мають обмеження на значення, які можуть бути присвоєні певним об’єктам.

Розподіл регістрів в мікропроцесах

Компілятор - це комп’ютерна програма, яка переводить одну комп’ютерну мову в іншу. Однією з технік компіляторної оптимізації для покращення часу виконання результуючого коду є розподіл регістрів.

У роботі [16] Цайтхофер та Весс описують використання розфарбування за списком для визначення правильного розподілу регістрів в обчислювальних процесах. Вони розглядають випадок з кількома функціональними одиницями: одні можна додавати і множити, інші — тільки додавати. Розфарбування за списком застосовують до складання розкладу інструкцій. Наприклад, якщо певна операція вимагає множення, то ця операція не може використовувати функціональну одиницю, що може тільки додавати. Так, кожній операції присвоюється свій список з функціональних одиниць, що можуть бути використані в цій операції і виникає проблема розфарбування за списком.

Друга проблема, що використовує розфарбування за списком, виникає при зберіганні проміжних значень в регістрах. Кожній операції присвоюється список з регістрів залежно від того до яких регістрів функціональні одиниці мають доступ. Наприклад, якщо операція вимагає додавання, а далі множення,

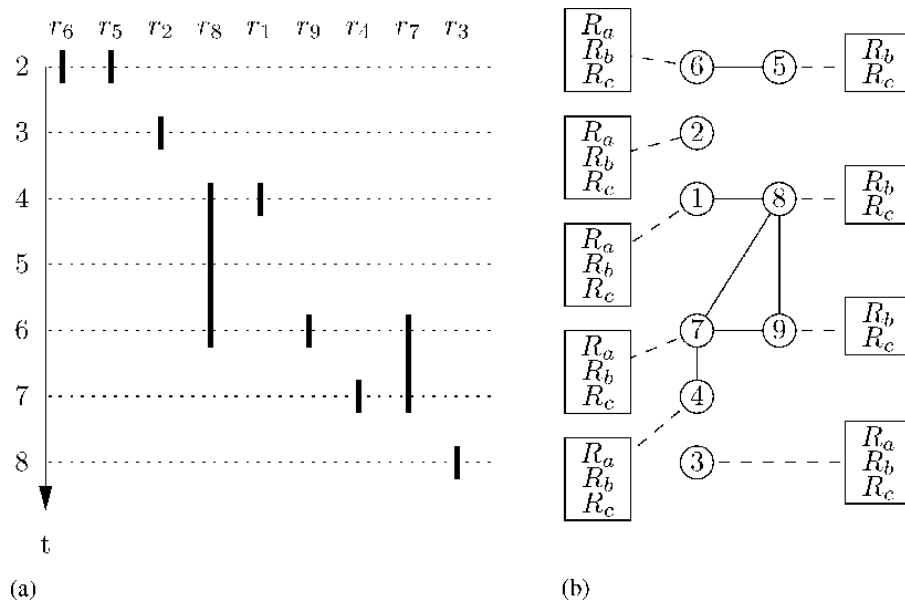


Рис. 44 [16]

то проміжне значення додавання не повинно зберігатися в регістрі, до якого функціональна одиниця множення немає доступу.

Розв'язання цих двох проблем приводить до правильного призначення регістрів та розкладу інструкцій для обчислювального процесу. Компілятор будує інтерференційний граф, де вершини відповідають регістрам, і з'єднуються ребрами, якщо вони потрібні в один момент часу. Приклад інтерференційного графу можна побачити на рис.44.

Розподіл каналів в бездротових мережах

У роботі [18] Рамачандран, Белдінг, Альмерот і Буддікот описували проблему втручання один до одного бездротових мереж. Кожний радіопристрій в мережі має обмежений набір частот, що дають змогу мати зв'язок. Такі обмеження накладає апаратне забезпечення. Радіопристрої, що знаходяться на певній відстані один від одного, не можуть працювати на одній частоті, коли передають по одному каналу. Таким чином виникає затримка в передачі по каналу. Щоб таких втручань не було, радіопристрої, що знаходяться на певній відстані, повинні бути з різною частотою. Так, проблема описується за допомогою розфарбування за списком для графа:

- вершини графа — бездротові радіопристрої в мережі
- вершина графа з'єднуються ребром, якщо відповідні їм радіопристрої знаходяться на певній відстані один від одного
- список доступних частот присвоюється до кожного радіопристрою, тобто до кожної вершини

На рис.45 кольори позначають частоту радіопристроїв в бездротовій мережі. Лінія з'єднує ті радіопристрої, що знаходяться на одній відстані і мають однакову частоту.

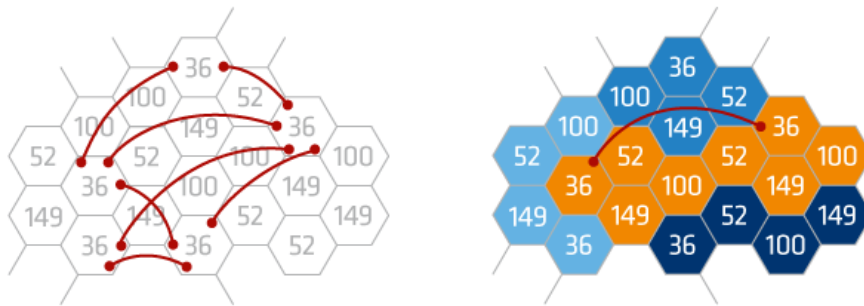


Рис. 45 [19]

Цифрові водяні знаки

Технологія цифрових водяних знаків разом з даними (наприклад, медіа-файли) передати зашифроване повідомлення, яке називається *водяний знак*. Зазвичай *водяний знак* використовуються для захисту авторських прав. Технологія є важливою для встановлення, хто поширює дані нелегальним чином, або для підтвердження прав на ці дані.

Повідомлення можна зашифрувати в графі і така техніка називається *QR-алгоритм*. Граф і його розфарбування, що використовується в алгоритмі шифрування, можна змінити так, щоб збільшилося хроматичне число. Приклад такого графу — інтерференційний граф. Зашифрувати повідомлення можна буде за допомогою розподілу регістрів в мікропроцесах.

Алгоритм був удосконалений у роботі [21]. Приклад шифрування значення **3** удосконаленим алгоритмом можна побачити на рис.46. Верхній граф

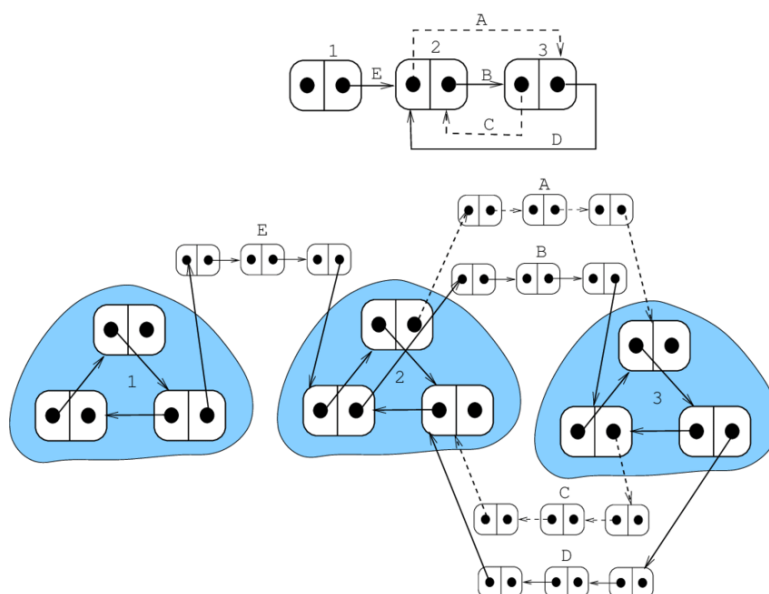


Рис. 46 [21]

показує графік кодування, нижній граф показує як це кодування стає більш стійким до хакерських атак, що намагаються розділити вузли. Таким чином кожний вузол графа перетворено у трьохцикловий вузол, а кожне ребро — в шлях довжиною три.

3 Висновки

В розділі 2.1 моєї кваліфікаційної роботи описана постановка і головне питання задачі Діниця.

В розділі 2.2 зроблено огляд найстаріших комбінаторних об'єктів - латинських квадратів, що є розв'язанням задачі Діниця. В підрозділі 2.2.1 описані основні означення про латинські квадрати. В підрозділі 2.2.2 розглянуті приклади спряжених латинських квадратів та представлення латинських квадратів. В підрозділі 2.2.3 представлена формула для підрахунку числа латинських квадратів та деякі вже знайдені числа.

В розділі 2.3 описано загальний випадок задачі Діниця.

В розділі 2.4 зроблено огляд основних означень з теорії графів, що допоможуть сформулювати задачу Діниця за допомогою графів. Означення хроматичного числа є недостатнім для розв'язання задачі Діниця, тому в підрозділі 2.4.1 описано хроматичне число розфарбування за списком. Також в підрозділі досліджено твердження про рівність хроматичного числа та хроматичного числа за списком, розглянуто граф S_n та приклади розфарбування за списком повних двочасткових графів.

В розділі 2.5 описано доведення нерівності $\chi_l(S_n) \leq n + 1$. Основні означення про орієнтовані графи та перша лема, що пояснює зв'язок між орієнтованим графом, що має ядро і розфарбуванням за списком, описані в підрозділі 2.5.1. В підрозділі 2.5.2 описано доведення леми Холла та основні означення, що стосуються паросполучень. Про паросполучення в двочастковому графі та друга лема, що доводить існування стійкого паросполучення, зробено огляд в підрозділі 2.5.3.

В розділі 2.6 представлено формулювання головної теореми Діниця і отримання результату, що $\chi_l(S_n) = n$. Доведення теореми використовує дві леми, що були доведені раніше.

В розділі 2.7 представлена практична частина. В підрозділі 2.7.1 описано знайдений на основі доведених теорем алгоритм знаходження розфарбування для задачі Діниця. В підрозділі 2.7.2 і підрозділі 2.7.3 представлено розв'язок і формулювання задач знаходження розфарбування графів S_3 та S_5 за алгоритмом, наведеним в 2.7.1. В підрозділі 2.7.4 і підрозділі 2.7.5 представлено застосування теореми 2 та алгоритму Гейла-Шеплі до графів S_3 та S_5 .

В розділі 2.8 зроблено огляд основних означень для реберного графа. Також описано формулювання гіпотези про рівність хроматичного числа та хроматичного числа за списком для реберного графа.

В розділі 2.9 представлено три прикладних застосування розфарбування за списком для графа: розподіл регістрів в мікропроцесах, розподіл каналів в бездротових мережах та цифрові водяні знаки.

4 Список літератури

1. *Айгнер М., Циглер Г.* Доказательства из Книги. Лучшие доказательства со времен Евклида до наших дней: Пер. с англ. — М.: Мир, 2006.
2. *M. Aigner, G.M. Ziegler.* Proofs from THE BOOK, Springer, 2003.
3. *Гладких О.Б., Белых О.Н.* Основные понятия теории графов: Учебное пособие. - Елец: ЕГУ им. И.А. Бунина, 2008.
4. *М. Э. Тужилин.* Латинские квадраты и их применение в криптографии, ПДМ, 2012, номер 3(17).
5. Stable marriage problem.
https://en.wikipedia.org/wiki/Stable_marriage_problem
6. List coloring.
https://en.wikipedia.org/wiki/List_coloring
7. *Jonathan L. Gross, Jay Yellen.* Handbook of Graph Theory, 2004.
8. *Д.В. Карнов.* Теория графов, 2017.
9. *Reinhard Diestel.* Graph Theory, Electronic Edition, Springer, 2000.
10. The Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel 2012.
<https://www.nobelprize.org/prizes/economic-sciences/2012/summary/>
11. Stable matching theory.
https://en.wikipedia.org/wiki/Stable_matching_theory
12. Latin square.
https://en.wikipedia.org/wiki/Latin_square
13. Mutually orthogonal Latin square.
https://en.wikipedia.org/wiki/Mutually_orthogonal_Latin_squares
14. *Jonathan Hirata.* Notes on Matching, 2013.

15. *Courtney L. Baber*. An Introduction to List Colorings of Graphs, 2009.
16. *T. Zeitlhofer and B. Wess*. List-coloring of interval graphs with application to registerassignment for heterogenous register-set architectures. *Signal Process.*, 83:1411–1425, 2003.
17. Register allocation.
https://en.wikipedia.org/wiki/Register_allocation
18. *K. Ramachandran, E. Belding, K. Almeroth, and M. Buddhikot*. Interference-aware channel assignment in multi-radio wireless mesh networks, 2006.
19. Frequency transmission.
<https://www.lancom-systems.com/technology/wi-fi-6/>
20. *Michelle Anne Lastrina*. List-coloring and sum-list-coloring problems on graphs, 2012.
21. *Collberg, Christian; Clark Thomborson, Gregg M. Townsend*. Dynamic Graph-Based Software Watermarking, 2004.
22. Line graph.
https://en.wikipedia.org/wiki/Line_graph
23. Lattice of stable matchings.
https://en.wikipedia.org/wiki/Lattice_of_stable_matchings