ЛЕКЦІЯ 14

Розфарбування графа

План лекції

1. Задачі розфарбування

Задачі розфарбування вершин або ребер графа займають важливе місце в теорії графів.

До задачі побудови розфарбування графа зводиться цілий ряд практичних задач.

Одна з областей – складання розкладів.

- розкладу для освітніх закладів;
- розкладу в спорті;
- планування зустрічей, зборів, інтерв'ю;
- розкладу транспорту, у тому числі — авіатранспорту;
- розкладу для комунальних служб;
- та інші.

2. Основні визначення

Нехай $G = (V, E)$ — скінченний граф, а k — деяке натуральне число.

Вершинне розфарбування. Довільну функцію виду $f : V \to N_k$, де $N_k = \{1, 2, ..., k\}$, називають *вершинним* k*-розфарбуванням*, або просто k*-розфарбуванням* графа G.

Правильне розфарбування. Розфарбуванням називають *правильним*, якщо кольори суміжних вершин не співпадають, тобто для всіх $(u, v) \in E$ справедливо $f(u) \neq f(v)$.

Розфарбований граф. Граф, для якого існує правильне k-розфарбування, називається *розфарбованим графом.*

Базовий принцип оптимізації розфарбування

Якщо функція f **не взаємно однозначна**, то при $|V| = k$ фактично може бути використано **менше,** ніж k кольорів.

Правильне розфарбування – це розбиття множини вершин

Правильне k-розфарбування можна розглядати як розбиття множини вершин V графа G на класи $V_1 \cup V_2 \cup ... \cup V_l = V$, де $l \leq k$, $V_i \neq \varnothing$, $i = 1, 2, ..., l$.

Кожний клас V_i — незалежна множина, а самі класи називаються *кольоровими класами.*

3. Хроматичне число

Визначення. Мінімальне число k, при якому існує правильне k-розфарбування графа G, називають *хроматичним числом* цього графа і позначають $X_p(G)$.

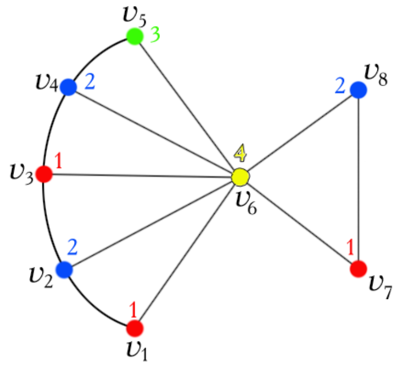
Визначення. Якщо $X_p(G) = k$, то граф G називають k*-хроматичним.* Тобто його вершини можна розфарбувати k різними кольорами так, що у будь-якого ребра інцидентні вершини матимуть різний колір.

Визначення. Правильна k-розфарбування графа G при $k = X_p(G)$ називається *мінімальної.*

Визначення. Хроматичне число незв'язного графа дорівнює максимальному з хроматичних чисел його компонент зв’язності.

Приклад.

Розглянемо граф G, зображений на рисунку, на якому показана одне із правильних k-розфарбувань. Натуральними числами 1,2,3,4 позначені фарби відповідних вершин.

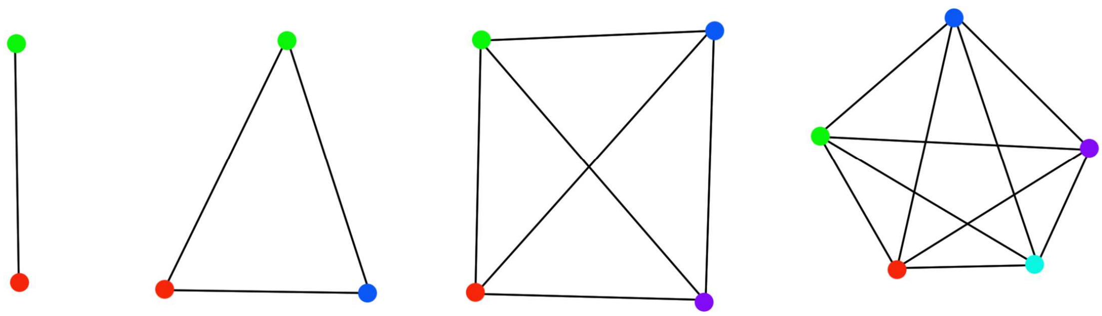


3.1. Хроматичні числа деяких графів

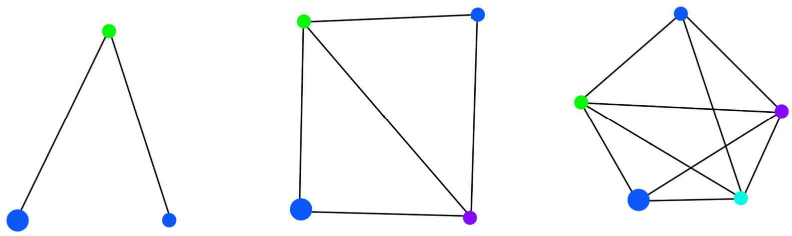
Для деяких простих графів неважко знайти хроматичні числа.

Приклади.

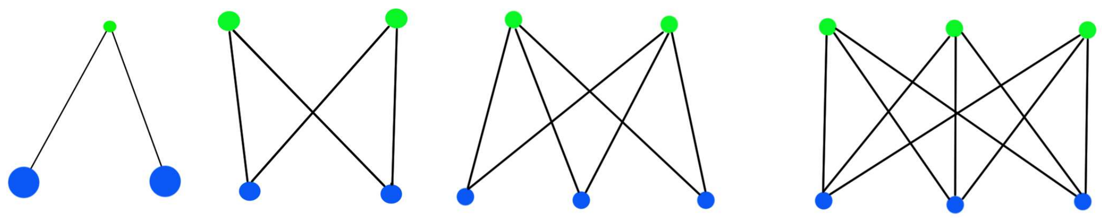
1. Повний граф K_n, що складається з n вершин, має хроматичне число $X_p\left(K_n\right) = n$



2. Повний граф $K_n - e$, який складається з n вершин з одним відсутнім ребром, має хроматичне число $X_p\left(K_n - e\right) = n - 1$



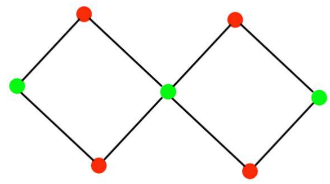
3. Повні дводольні графи $K_{m,n}$, що складаються з долей $\left|A\right| = m$ і $\left|B\right| = n$, мають хроматичне число $X_p\left(K_{m,n}\right) = 2$



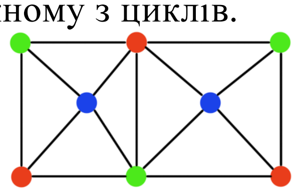
Теорема. Непустий граф є **біхроматичним** тоді й тільки тоді, коли він не має циклів непарної довжини.

1-хроматичний граф – порожній граф.

2-хроматичний граф — дводольний непустий граф, 2-хроматичні графи, як правило, називають *біхроматичними.*



3-хроматичний граф — циклічний граф з непарним числом вершин у кожному з циклів.

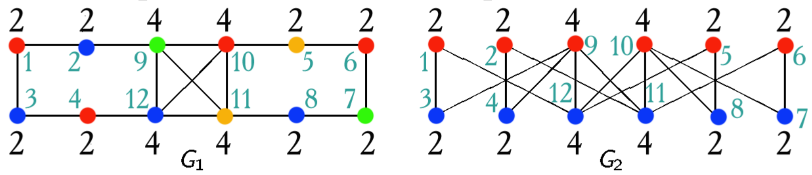


Якщо граф має n вершин, то його хроматичне число не перевищує n

Якщо граф має підграф K_m, то його хроматичне число не менше m.

4. Хроматичне число й стандартні характеристики

У загальному випадку хроматичне число графа не можна обчислити, знаючи тільки його стандартні числові характеристики: *число вершин, ребер,*
компонент зв'язності, розподіл степенів вершин.



Розглянемо графи G_1 й G_2. Кожний з них має 12 вершин, у тому числі 4 вершини зі степенем 4 і вісім вершин зі степенем 2, шістнадцять ребер, один компонент зв'язності. Але, як видно з рисунка, $x(G_1) = 4$, а $x(G_2) = 2$, оскільки G_1 містить у якості підграфа граф K_4.

Оскільки граф G_2 — дводольний, маємо $x(G_2) = 2$.

Тому надалі мова йтиме про оцінки, а не про точні значення хроматичного числа.

5. Хроматичне число і щільність графа

Під нижніми оцінками хроматичного числа будемо розуміти нерівності виду $X\left(G\right) \geq c$, де c — деяка константа, що обчислюється на графі G.

Верхня оцінка хроматичного числа — це нерівності виду $X(G) \le c$, де c має той же зміст.

Визначення. Максимальне число вершин, що породжують повний підграф у графі G, називають ***щільністю*** G і позначають через $\omega(G)$.

Визначення. Повний підграф деякого графа G - це підграф, що складається з попарно суміжних вершин.

Перша нижня оцінка може застосовуватися у випадку якщо в якості підграфа деякого графа присутній повний підграф.

Перша нижня оцінка

Для довільного графа G справедлива нерівність $X(G) \ge \omega(G)$.

6. Хроматичне число і число незалежності графа

Визначення. Будь-яку множину попарно несуміжних вершин графа G називають *незалежною множиною.*

Визначення. Максимальне число вершин у незалежній множині називають *числом незалежності* графа G й позначають через $\beta(G)$.

Число незалежності графа — це поняття, протилежне за змістом поняттю щільності графа. Якщо G — звичайний граф, а $\overline{G}$ — його доповнення, то $\beta(G) = \omega(\overline{G})$.

Друга нижня оцінка

Для довільного графа G справедлива нерівність

$$X(G) \ge \frac{n(G)}{\beta(G)}$$

7. Третя нижня оцінка хроматичного числа

Існують **нижні оцінки** хроматичного числа, які використовують тільки ті характеристики графа, що легко обчислюються. Наведемо без доведення одну з них.

Третя нижня оцінка

Якщо G – звичайний граф і

$n = n(G)$ – кількість вершин графа G,

$m = m(G)$ – кількість ребер графа G,

то хроматичне число $X(G) \ge \dfrac{n^2}{n^2 - 2m}$.

Легко зрозуміти, що в повному графі
(як і в будь-якому звичайному графі)
подвоєне число ребер менше квадрата числа вершин, і тому число, що стоїть в знаменнику в правій частині нерівності, завжди додатне.

Як видне з описаних вище результатів, **задачі визначення хроматичного числа** графа й побудови мінімального розфарбування довільного графа **досить складні**, а ефективні алгоритми їх розв'язку невідомі. Розглянемо простий алгоритм побудови правильного розфарбування, який у деяких випадках дає розфарбування, близькі до мінімальних.

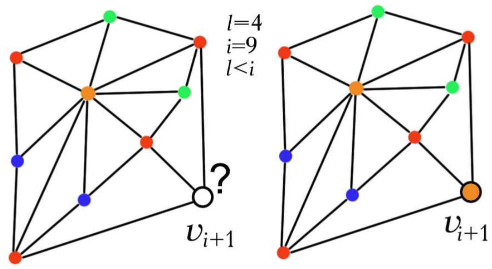
8. Алгоритм послідовного розфарбування

1. Якщо вершини $v_1, v_2, ..., v_i$ розфарбовані l кольорами $1, 2, ..., l$; $l \leq i$, то новій довільно взятій вершині v_{i+1} припишемо мінімальний колір, не використаний при розфарбуванні суміжних з нею вершин.

Розфарбування, до якого приводить описаний алгоритм, називають *послідовним*.

9. Алгоритм послідовного розфарбування з упорядкуванням множини вершин

1. Упорядкувати вершини по незростанню степеня.
2. Вибрати колір фарбування 1.
3. Розфарбувати першу вершину в колір 1.
4. Поки не пофарбовані всі вершини, повторювати п.4.1.-4.2.:
4.1. Розфарбувати в обраний колір усяку вершину, яка не суміжна з іншою вершиною, уже пофарбованою в цей колір.
4.2. Вибрати наступний колір.
5. Повернутися до першої в списку нерозфарбованої вершини.



Число використаних квітів буде тоді наближеним значенням хроматичного числа графа.

10. Модифікація алгоритму послідовного розфарбування

Визначення. Відносний степінь – це степінь незабарвлених вершин у незабарвленому підграфі початкового графа.

Визначення. Двокроковий відносний степінь – сума відносних степенів суміжних вершин у нерозфарбованому підграфі**.**

Проста модифікація описаної вище евристичної процедури полягає в переупорядкуванні нерозфарбованих вершин по незростанню їх відносних степенів.
У даній модифікації передбачалося, що якщо дві вершини мають однакові степені, то порядок таких вершин випадковий. Їх можна впорядкувати за двокроковими степенями. Двокроковий степінь визначимо як суму відносних ступенів інцидентних вершин. Аналогічно можна продовжувати далі.

11. Рекурсивна процедура послідовного розфарбування

1. Фіксуємо порядок обходу вершин.
2. Ідемо по суміжних вершинах, використовуючи такий найменший колір, який не викличе конфліктів.
3. Якщо на черговому кроці колір вибрати не виходить, то «відкочуємось» до попередньої вершини й вибираємо для неї наступний колір, який не викличе конфліктів.

```
procedure visit(i: integer);
begin
 if i = n + 1 then Print else
 begin
  for c := color[i]+1 to k do//k-кількість кольорів
  if (знайдено неконфліктний колір c) then
  begin color[i] := c; visit(i + 1); end else
  visit(i);
 end;
end;
```

12. «Жадібний» алгоритм розфарбування

Нехай дано зв'язний граф $G\left(V,E\right)$.

1. Задамо множину $monochrom := \varnothing$, куди будемо записувати всі вершини, які можна розфарбувати одним кольором.

2. Переглядаємо всі вершини й виконуємо наступний «жадібний» алгоритм

Procedure Greedy
For (для кожної незафарбованої вершини $v \in V$) **do**
If v *не суміжна з вершинами з* $monochrom$ **then**
begin
$color\left(v\right) :=$ колір;
$monochrom := monochrom \cup \left\{v\right\}$
end.

13. Розфарбування графа методом А.П. Єршова

Андрій Петрович Єршов (1931-1988 рр.), найбільший учений в області теоретичного програмування, вніс великий вклад у розвиток інформатики в нашій країні. Їм створений алгоритм розфарбування графа, що виділяється оригінальної евристичною ідеєю.

Уведемо ряд визначень.

Для даної вершини $v \in V$ графа $G(V, E)$ назвемо всі суміжні з нею вершини околицею 1-го порядку — $R_1(v)$.

Усі вершини, що перебувають від v на відстані два, назвемо околицею 2-го порядку — $R_2(v)$.

Граф $G(V, E)$, у якого для вершини $v \in V$ всі інші вершини належать околиці $R_1(v)$ назвемо граф-зіркою щодо вершини v.

13.1. Ідея алгоритму

Фарбування у фарбу α вершини v утворює навколо неї в $R_1(v)$ «мертву зону» для фарби α. Очевидно, при мінімальнім розфарбуванні на кожну фарбу повинне доводитися найбільше число вершин графа. Для цього необхідно, щоб мертві зони хоча б частково перекривалися між собою. Перекриття мертвих зон двох несуміжних вершин v_1 і v_2 досягається тільки тоді, коли одна з них перебуває в околиці $R_2(v_1)$ від інший, $v_2 \in R_2(v_1)$.

Таким чином, суть алгоритму полягає в тому, щоб на черговому кроці вибрати для розфарбування фарбою α вершину $v_2 \in R_2(v_1)$. Цей процес повторювати доти, поки фарбою α не будуть пофарбовані всі можливі вершини графа.

Графічно фарбування вершин v_1 і v_2 однієї фарбою можна відобразити як «склеювання» цих вершин.

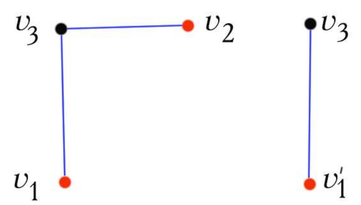


Рис. 14.1. Приклад об'єднання двох вершин: $v_1' := v_1 \cup v_2$

При цьому зменшується на одиницю число вершин у графі G й зменшується число ребер.

Алгоритм

1.Покласти $i := 0$.

2.Вибрати в графі G довільну незабарвлену вершину v.

3.Покласти $i := i + 1$.

4.Офарбити вершину v у фарбу i.

5.Офарблювати у фарбу i незабарвлені вершини графа G, вибираючи їх з $R_2(v)$, поки граф не перетвориться в граф-зірку відносно v .

6.Перевірити, чи залишилися незабарвлені вершини в графі G. Якщо так, то перейти до п. 2, інакше - до п. 7.

7.Отриманий повний граф K_i. Хроматичне число графа $X(K_i) = i$.

Кінець алгоритму.

Як випливає из алгоритму, після того, як у першу обрану вершину стягнуті всі можливі й граф перетворився в граф-зірку щодо цієї вершини, вибирається довільним образом друга вершина й процес повторюється доти, поки граф не перетвориться в повний.

Повний граф - це граф-зірка щодо будь-якої вершини. Хроматичне число повного графа дорівнює числу його вершин.

13.2. Приклад розфарбування методом А.П. Єршова

На мал.14.2 зображений граф G, на прикладі якого будемо розглядати роботу евристичного алгоритму Єршова.

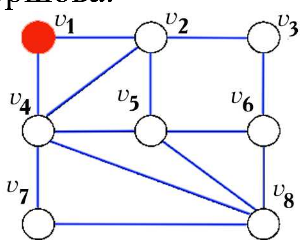


Рис.14.2. Вихідний граф G

Виберемо довільну вершину, наприклад, v_1. Околиця 2-го порядку $R_2(v_1) = \{v_3, v_5, v_7, v_8\}$. Склеїмо вершину v_1 наприклад, з вершиною v_3: $v_1' = v_1 \cup v_3$. Одержимо граф G_1 зображений на мал. 14.3.

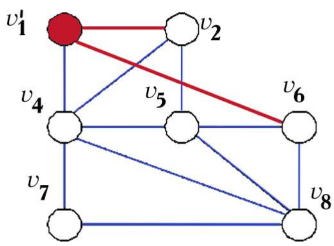


Рис.14.3. Граф G_1. Склеєні вершини v_1 й v_3

Розглянемо тепер граф G_1. Околиця другого порядку вершини v_1' визначається множиною $R_2(v_1') = \{v_5.v_7, v_8\}$. Склеїмо вершину v_1', наприклад, з вершиною v_5: $v_1'' := v_1' \cup v_5$. Одержимо граф G_2, зображений на мал. 14.4.

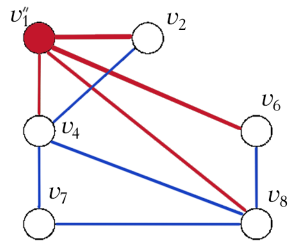


Рис.14.4. Граф G_2. Склеєні вершини v_1' й v_5

У графі G_2 визначимо околиця другого порядку для вершини v_1'' : $R_2\left(v_1''\right)=\left\{v_7\right\}$. Склеїмо вершину v_1'' з вершиною v_7: $v_1''' = v_1'' \cup v_7$. Одержимо граф G_3, зображений на мал. 14.5 граф, що є, езіркою щодо вершини v_1'''.

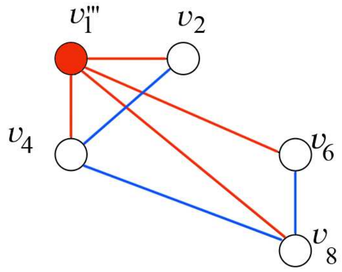


Рис.14.5. Граф G_3. Склеєні вершини v_1'' й v_7

У графі G_3 виберемо, наприклад, вершину v_2 для фарбування другою фарбою. Околиця 2-го порядку $R_2\left(v_2\right)=\left\{v_6, v_8\right\}$. Склеїмо вершину v_2, наприклад, з вершиною v_6: $v_1''' = v_1'' \cup v_6$. Одержимо граф G_4, зображений на мал.14.6.

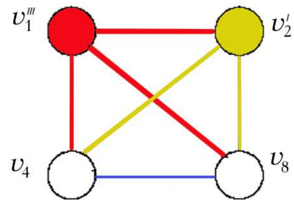


Рис.14.6. Граф G_4 еквівалентний K_4. Склеєні вершини v_1'' й v_6

Граф G_4 є повним четырехвершинным графом ДО4. Отже, граф G_4 на мал.14.6 розфарбовується в чотири кольори. У перший (червоний) колір розфарбовується вершина v_1 й склеєні з нею вершини: v_3, v_5 і v_7. У другий (жовтий) колір розфарбовується вершина v_2 й склеєна з нею вершина v_6. У третій (зелений) колір розфарбовується вершина v_4 й у четвертий (білий) колір розфарбовується вершина v_8

У підсумку одержуємо правильно розфарбований граф, показаний на мал.14.7.

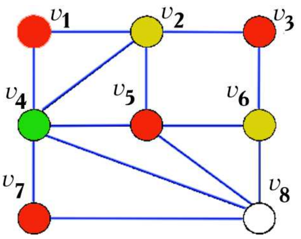
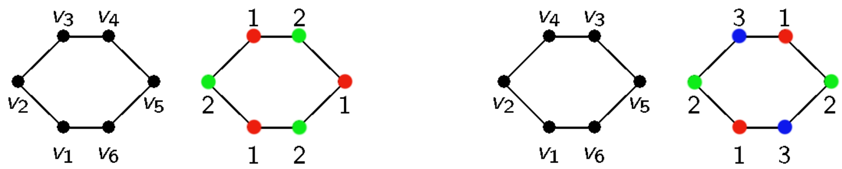


Рис.14.7.Граф G , розфарбований за допомогою алгоритму розфарбування А.П.Єршова

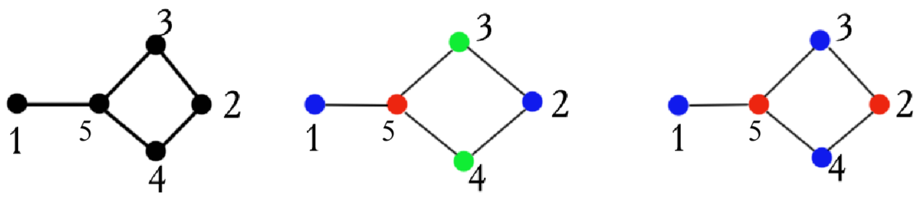
14. Результати роботи алгоритмів послідовного розфарбування

Отримане розфарбування завжди правильне, але не завжди оптимальне навіть для простих графів.



Воно суттєво залежить від того, у якому порядку вибираються вершини. На першому рисунку вийшов оптимальний результат (2 фарби), а на другому рисунку використано більше фарб.

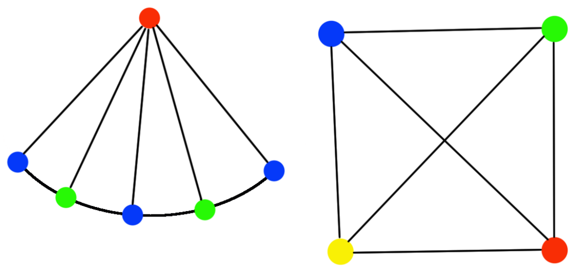
Нехай зафарбуємо вершину 1 у синій колір, а потім, пропустивши вершину 2, зафарбуємо в синій колір вершини 3 і 4. Тоді можна одержати 2 фарби.



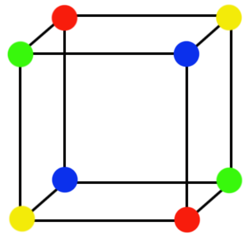
Але "жадібний" алгоритм, ґрунтуючись на нумерації вершин, зафарбує в синій колір вершини 1 і 2, для розфарбування графа тепер потрібно 4 фарби.

Тому актуальна верхня оцінка хроматичного числа.

Теорема 2. Для всякого графа G має місце нерівність $X_p\left(G\right) \le r+1$, де $r = \max\limits_{v \in V}\left(\deg\left(v\right)\right)$.



Наслідок. Усякий кубічний граф розфарбовується за допомогою чотирьох фарб.



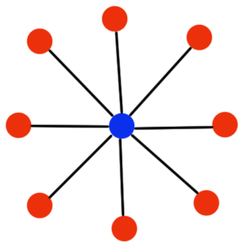
Більш загальний результат, ніж розглянута теорема, дає така теорема.

15. Теорема Брукса. *Якщо G — зв'язний неповний граф* і

$$r \geq 3,\ де\ \ r = \max_{v \in V}\left(\deg\left(v\right)\right)\ то\ \ X_p\left(G\right) \leq r\ .$$

Хоча обидві теореми й дають певну інформацію про хроматичне число графа, але їх оцінки досить неточні.

Дійсно, **зоряний граф** K_{1n}, який згідно з теоремою Брукса **розфарбовується** n **фарбами**, насправді є **біхроматичним**.



Ця ситуація значно спрощується, якщо обмежитися ***планарними графами***. У цьому випадку легко довести такий досить загальний і важливий факт.

16. Історично послідовно доведені теореми

Теорема про шість фарби. Для всякого планарного (ізоморфного плоскому (у якому ребра перетинаються лише у вершинах)) графа *G* має місце нерівність $X_p\left(G\right) \leq 6$.

Більш детальний аналіз шляхів зниження верхньої границі хроматичного числа приводить до так званої *теореми про п'ять фарби.*

Теорема про п'ять фарби. Для всякого планарного графа *G* має місце нерівність $X_p\left(G\right) \leq 5$.

Теорема про чотири фарбах. Кожний планарный граф без петель і кратних ребер є не більш ніж 4-хроматичним.

Проблема чотирьох фарб залишалася невирішеною протягом багатьох років. Стверджується, що ця теорема була доведена за допомогою певних міркувань і комп'ютерної програми в 1976 році (**Kenneth Appel and Wolfgang Haken. Every Planar Map is Four Colorable. Contemporary Mathematics 98, American Mathematical Society, 1980**).

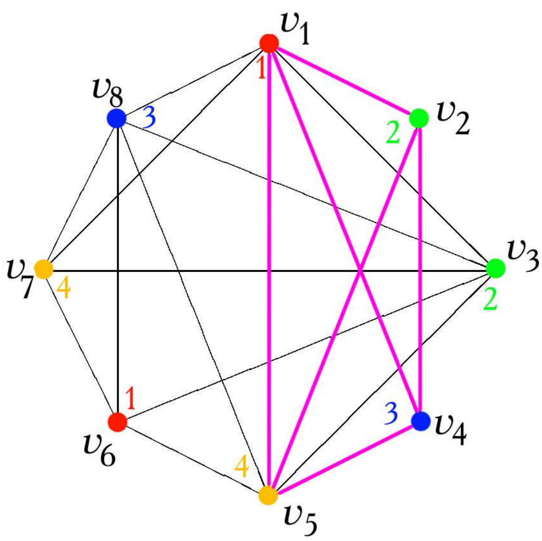
17. ЗАДАЧА РОЗПОДІЛУ УСТАТКУВАННЯ

На підприємстві планують виконати 8 робіт $v_1, v_2, \ldots, v_8$. Для виконання цих робіт необхідні механізми $a_1, a_2, \ldots, a_6$. Використання механізмів для кожної з робіт визначається наступною таблицею:

Механізм	Робота						
	v_1	v_2	v_3	v_4	v_5	v_6	v_7
a_1	+		+				+
a_2		+		+			
a_3			+			+	+
a_4	+	+		+	+		
a_5			+		+		
a_6					+	+	

Жоден з механізмів не може бути використаний одночасно на двох роботах. Виконання кожної роботи займає 1 годину. Як розподілити механізми, щоб сумарний час виконання всіх робіт був мінімальним і який цей час?

Розв'язок. Розглянемо граф G, вершинами якого є плановані роботи $v_1, v_2, \ldots, v_8$, а ребра з'єднують роботи, у яких бере участь хоча б один загальний механізм (і які, із цієї причини, не можна проводити одночасно).



Вершини v_1, v_2, v_4, v_5 породжують підграф графа G, ізоморфний K_4. Отже, $X\{G\} \geq 4$. Отже, $X(G)$.

Таким чином, усі роботи можна виконати за 4 години.

Для цього, відповідно до знайденого розфарбування графа G, треба за 1-у годину виконувати роботи v_1 і v_6, за 2-у годину — роботи v_2 і v_3, за 3-ю годину — роботи v_4 і v_8, за 4-у годину — роботи v_5 й v_7.

Механізм	Робота							
	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
a_1	+		+				+	+
a_2		+		+				
a_3			+			+	+	
a_4	+	+		+	+			
a_5			+		+			+
a_6					+	+		+

18. Задача складання розкладу

Умова задачі

Потрібно прочитати лекції з чотирьом предметів двом групам студентів. Деякі з лекцій не можуть читатися одночасно.

Існує ряд причин, по яких це неможливо:

1. Лекцію по даному предмету в різних групах читає той самий лектор.
2. Дві лекції однієї й тій же групі одночасно читатися не можуть.
3. Різні лекції повинні проходити в тому самому приміщенні.

Потрібно скласти розклад так, щоб читання всіх лекцій зайняв мінімально можливий час
(у якості «одиниці часу» у даній задачі природно розглядати одну пару).

Конкретизуємо постановку задачі

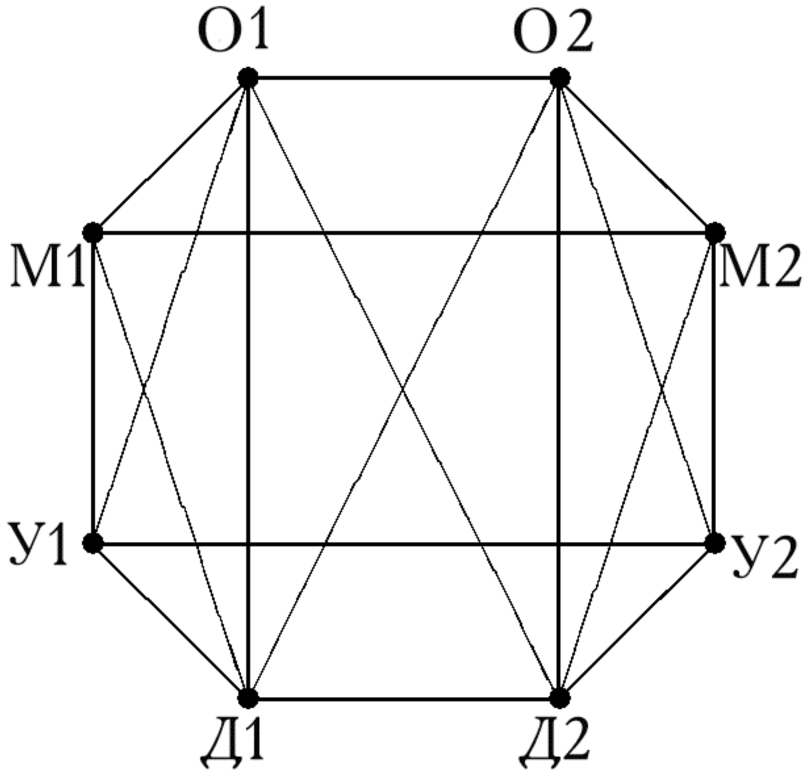
1. Будемо розглядати дві групи студентів: 1 і 2.

2. предмети, що читаються:
- обчислювальні методи – читає викладач X,
- дискретна математика – читає викладач X,
- математичний аналіз – читає викладач Y,
- українська мова – читає викладач Z.

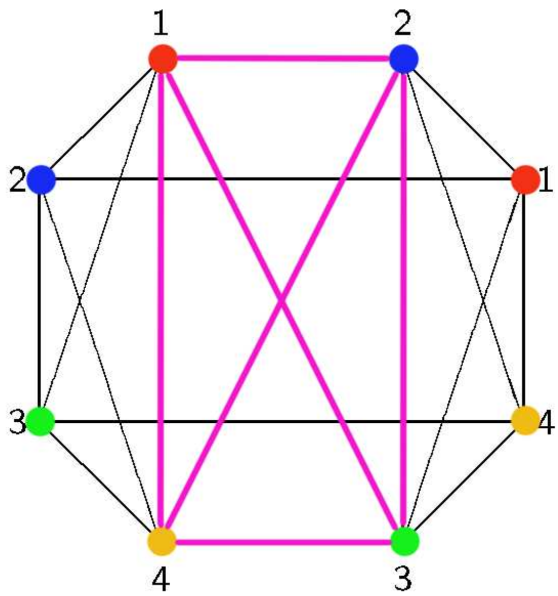
Знайти мінімальне число пар, у які можна «укласти» усі заняття, і скласти відповідний розклад.

Тобто створити, максимально щільний розклад

Розв'язок. Побудуємо граф з вершинами О1, О2, Д1, Д2, М1, М2, У1 і У2 (буква відповідає предмету, а цифра — номеру групи).
З'єднаємо ребрами вершини, відповідні до пар, які не можна проводити одночасно.
Одержимо граф, показаний на рисунку:



Вершини О1, О2, Д1 і Д2 цього графа породжують у ньому підграф, ізоморфний графу K_4. Отже, хроматичне число нашого графа не менше 4. На рисунку зазначене правильне розфарбування нашого графа в 4 фарби.



Отже, хроматичне число графа дорівнює 4, тобто всі заняття можна провести за 4 пари. Відповідний розклад зазначений у таблиці на наступному слайді.

	Група 1	Груп
1 пара	Обч. методи	Матем. аналіз
2 пара	Матем. аналіз	Обч. методи
3 пара	Українська мова	Дискр. матем
4 пара	Дискр. матем.	Українська мова