Хроматичне число

Розфарбуємо кожну вершину (відповідно, кожне ребро) графа G в які-небудь кольори.

Визначення. Розфарбування вершин (ребер) графа називається *правильним*, якщо ніякі дві суміжні вершини (два суміжних ребра) не пофарбовані в один й той самий колір.

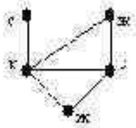
Означення. Граф G називається p -*хроматичним*, якщо він допускає правильне вершинне фарбування у p фарб.

Правильне розфарбування вершин звичайно називають просто розфарбуванням графа.

Визначення. Найменше число кольорів $p = \gamma(G)$, якими можна розфарбувати граф, називається *хроматичним числом* цього графа.

Хроматичне число

Наприклад, граф, показаний на малюнку, є 5-хроматичним, 4-хроматичним, 3-хроматичним.



Його хроматичне число 3.

Теорема 1 (Кеніг). Для того, щоб граф був біхроматичним, необхідно й достатньо, щоб він не мав циклів непарної довжини.

Достатність. Нехай G – граф без циклів непарної довжини, x – його вершина. Офарбимо її в червоні кольори. Всі суміжні з нею вершини офарбимо в сині кольори. Всі вершини, суміжні з пофарбованими синім, офарбимо в червоні кольори й т.д. при цьому ніяка вершина y не може виявитися й червоною, і синьою, тому що це значило б, що існує два ланцюги, що з'єднують x й y, з яких одна – парної довжини, а інша – непарної (якщо об'єднаємо їх у цикл, то одержимо цикл непарної довжини, протиріччя).

Необхідність. Нехай G – біхроматичний граф. Розфарбуємо його вершини в червоний і синій кольори так, щоб ніякі дві суміжні вершини не були пофарбовані в один колір. Нехай z – деякий цикл графа G. При його обході червоні й сині вершини будуть чергуватися. Отже, загальне число вершин циклу, а значить і ребер, парно.

Нижче приведемо один зі способів знаходження хроматичного числа.

1. Довільним образом правильно розфарбуємо граф G кольорами $1,2,\ldots,p$.
2. Вибираємо деякі кольори j в якості «критичного» і намагаємося виключити його. Нехай x – деяка вершина кольорів j, G^{ik} – підграф графа G, породжений вершинами кольорів i,k, відмінних від j, $G_1{}^{ik}, G_2{}^{ik},\ldots$ – компоненти зв'язности графа $G_1{}^{ik}$; $M_s{}^x$ – множина всіх суміжних з x вершин у компоненті $G_s{}^{ik}$. Якщо не всі множини $M_s{}^x$ одноколірні, то якщо буде потреба поміняємо в $G_s{}^{ik}$ між собою кольори i,k так, щоб всі $M_s{}^x$ були пофарбовані в один колір, наприклад в i. Після цього вершину x перефарбуємо в кольори k. Розфарбування вершин графа при цьому залишається правильним.

3. Якщо залишилися вершини, що мають обрані раніше критичні кольори, то повторюємо крок 2 для цих кольорів. Якщо ні, то вибираємо в якості критичного інші кольори. Повторюємо крок 2 доти, поки це можливо.

Приклад. На мал. 1 наведений граф, пофарбований в 4 кольори (їм відповідають цифри 1,2,3,4).

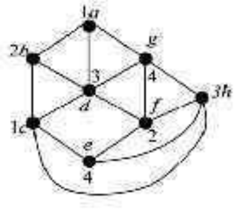


рис. 1

Виберемо в якості критичного кольору 1. Їм пофарбовані вершини a й c. Розглянемо компоненти зв'язності $G_1{}^{24}$ й $G_2{}^{24}$. Перша з них породжена вершинами e, f й g, друга вершиною b. У кожній компоненті тільки одна вершина суміжна з вершиною a, $M_1{}^a = \{g_4\}, M_2{}^a = \{b_2\}$. Міняємо в компоненті $G_1{}^{24}$ кольори 2 й 4 між собою. У результаті одержимо розфарбування, показане на мал. 2

Хроматичне число

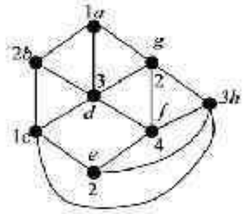


Рис.2

Тепер вершини a й c офарбимо в кольори 4 (мал. 3).

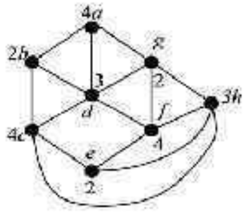


Рис.3

Граф виявляється розфарбованим в 3 кольори. Оскільки він містить цикли непарної довжини, тому меншим числом кольорів його розфарбувати не можна. Отже, його хроматичне число 3.

Хроматичне число

Помітимо, що описаний вище спосіб не завжди приводить до відшукання хроматичного числа графа. У книзі Крістофідеса "Теорія графів" наведені й інші способи відшукання хроматичного числа графа, у тому числі й позбавлені зазначеного недоліку - вони завжди приводять до шуканого результату.

Хроматичний клас

Найменше число фарб $p = \gamma'(G)$, достатніх для правильного реберного розфарбування графа G , називається *хроматичним класом* (іноді – реберним хроматичним числом).

Нехай G – найбільша зі ступенів вершин графа (тут при підрахунку ступенів будь-яка петля вважається 1 раз). Тоді, мабуть $k \leq \gamma'(G)$, . К. Шеннону належить теорема 2, доказ якої ми не будемо приводити.

Теорема 2. Хроматичний клас графа G не перевершує $\left[\frac{3}{2} \cdot k\right]$ (де $[x]$ – ціла частина числа).

Легко бачити, що поліпшити оцінку Шеннона без залучення додаткової інформації про граф не можна. Дійсно, вершини a, b, c графа, зображеного на мал.4, мають ступеня $\rho(a) = \rho(b) = k = 9$ й $\rho(c) = 2 \cdot \left[\frac{k}{2}\right] = 8$, а для правильного розфарбування його потрібно $\left[\frac{3}{2} \cdot k\right] = 13$ кольорів.

Хроматичний клас

Залучення ж додаткової характеристики графа – найбільшої кількості ребер між суміжними вершинами дозволило радянському математикові В. Г. Візінгу значно поліпшити цю оцінку.

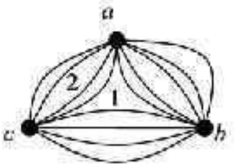


Рис.4

Теорема 3. (Візінг). Хроматичний клас графа G задовольняє нерівності

$$\gamma'(G) \leq k + p$$

Визначення. Граф G' називається *реберним графом* графа G, якщо він побудований у такий спосіб (мал.5):

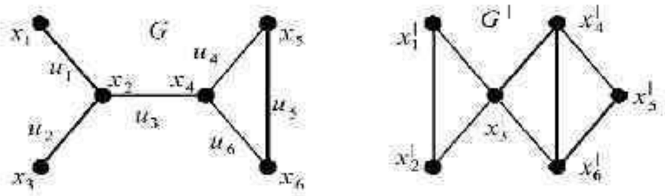


Рис.5

Хроматичний клас

1. кожному ребру u_k графа G ставиться у відповідність одна й тільки одна вершина x'_k графа G ;
2. вершини x'_k й x'_l графа G суміжні тоді й тільки тоді, коли суміжні відповідні їм ребра u_k й u_l графа G . Хроматичний клас графа G дорівнює хроматичному числу його реберного графа G'.

Поняття хроматичного класу графа використовується на практиці для побудови монтажних схем: проведення, що підходять до однієї крапки, щоб уникнути плутанини зручно офарблювати в різні кольори. Природно, виникає завдання про хроматичний клас графа монтажної схеми.

Число внутрішньої стійкості

Нехай $G = G(X, \Gamma)$ – граф без петель і кратних ребер.

Визначення. Підмножина вершин $S \subseteq X$ називається *внутрішньо стійким*, якщо ніякі дві вершини не суміжні. Нехай σ – сімейство всіх внутрішньо стійких множин графа G .

Визначення. Величина $\alpha(G) = \max\limits_{S \in \sigma}|s|$ називається *числом внутрішньої стійкості* графа G .

Приклад 1 (Гаусс). Чи можна на шахівниці розставити 8 ферзів так, щоб жоден з них не перебував під ударом якого-небудь іншого? Побудуємо граф G з 64 вершинами – клітками шахівниці; вершини x, y будемо вважати суміжними , якщо вони перебувають на одній вертикалі, горизонталі або діагоналі. Завдання зводиться до знаходження внутрішньо стійких множин у графі G таких, що $|S| = 8$ – вони виявляються найбільшими в G. Пронумерувавши горизонталі й вертикалі шахівниці, можемо записати розв'язання задачі у вигляді $(\gamma_1, \gamma_2, \gamma_3, \gamma_4, \gamma_5, \gamma_6, \gamma_7, \gamma_8)$, де (γ_k, k) – координати кожного з ферзів. От 12 рішень (усього завдання має 92 розв'язання):

Число внутрішньої стійкості

(7 2 6 3 1 4 8 5)	(6 1 5 2 8 3 7 4)
(5 8 4 1 7 2 6 3)	(3 5 8 4 1 7 2 6)
(4 6 1 5 2 8 3 7)	(5 7 2 6 3 1 4 8)
(1 6 8 3 7 4 2 5)	(5 7 2 6 3 1 8 4)
(4 8 1 5 7 2 6 3)	(5 1 4 6 8 2 7 3)
(4 2 7 5 1 8 6 3)	(3 5 2 8 1 7 4 6)

- **Приклад 2.** *Клікою* графа $G = G(X, \Gamma)$ називається така підмножина $C \subseteq X$, що породжений цією підмножиною підграф G є повним. Підмножина $C \subseteq X$ тоді й тільки тоді є клікою графа G, коли воно внутрішньо стійко в його доповненні$\overline{G}$.

Теорема 4. Хроматичне число $\gamma(G)$ й число внутрішньої стійкості $\alpha(G)$ зв'язані нерівністю

$$\alpha(G) \cdot \gamma(G) \geq |X|.$$

Доказ. Розфарбуємо (правильно) вершини графа G у $\gamma(G)$ кольорів. Кожна множина вершин, пофарбованих у той самий колір, внутрішньо стійка й, отже, $|X_i| \leq \alpha(G)$. Виходить, $|X| = \sum|X_i| \leq \alpha(G)\gamma(G)$.

Число внутрішньої стійкості

- ***Визначення.*** Відображення множини в себе називається зберігаючим, якщо для будь-якої пари різних несуміжних вершин вершини $\sigma(x)$ й $\sigma(y)$ також несуміжні й різні.

Твердження. Зберігаюче відображення σ переводить усяку внутрішньо стійку множину S у внутрішньо стійку множину.

У додатках зустрічаються різні визначення операцій над графами. От одне з них. Нехай $G = G(X, \Gamma)$ і $H = H(Y, \Lambda)$ графи без кратних ребер.

Добутком їх називається граф , множиною вершин якого є Декартів добуток , а вершини суміжні, якщо

1) x суміжна з x' в G , y суміжна з y' в H , або

2) x суміжна з x', $y = y'$ (або навпаки).

Природно визначається й квадрат $G^2 = G \times G$ і ступінь G^2 .

Теорема 5 (Шеннон). 1. Для будь-яких графів $G = G(X, \Gamma)$ й $H = H(Y, \Lambda)$, $\alpha(G) \cdot \gamma(H) \leq \alpha(G \times H)$.

2. Якщо для графа G існує зберігаюче відображення σ таке, що множина $\sigma(X)$ внутрішньо стійка, то $\alpha(G) \cdot \gamma(H) = \alpha(G \times H)$.

Число внутрішньої стійкості

Доведення.

1. Якщо S й T – найбільші внутрішньо стійкі множини для G й H, то $S \times T$ внутрішньо стійке й $\alpha(G \times H) \geq |S \times T| = |S| \cdot |T| = \alpha(G)\alpha(H)$.

2. Нехай тепер $\sigma-$ зберігаюче відображення для графа G й множина $\sigma(X)$ внутрішньо стійка. Для довільних $x \in X, y \in Y$ позначимо $\varphi(x, y) = (\sigma(x), y)$. Нехай S_0 – найбільша внутрішньо стійка множина графа $G \times H$. Легко перевірити, що також внутрішньо стійке й $|\varphi(S_0)| = |S_0|$. Розподілимо елементи з $\varphi(S_0)$ по k класам залежно від елемента x в парі (x, y). Будь-який клас містить не більш ніж $\alpha(H)$ елементів. Отже, $\alpha(G \times H) = |\varphi(S_0)| \leq k \cdot \alpha(H) \Longrightarrow \alpha(G \times H) \leq \alpha(G) \cdot \alpha(H)$.

Звідси й з першого твердження теореми випливає доказувана рівність.

Приклад 3. (Шеннон). Завдання про інформаційну ємність множини сигналів.

Число внутрішньої стійкості

Передавач може передавати п'ять сигналів a, b, c, d, e; при прийомі сигнала a може бути витлумачений як a або b, сигнал b – як b або c, c – як c або d, d – як d або e, e – як e або a . Яка найбільша кількість сигналів можна прийняти, не боячись поплутати їхній один з одним?

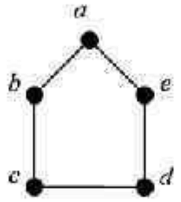


рис. 6

Завдання зводиться до знаходження найбільшої внутрішньо стійкої множини S графа на мал.6. Очевидно, що тут $\alpha(G) = 2$ і можна вибрати $S = \{a, c\}$. Якщо замість однобуквених сигналів користуватися словами довжини 2, те $[\alpha(G)]^2 = 4$, слова aa, ac, ca, cc – не можна поплутати. Можна однак скласти такий же надійний, але більше багатий код:aa, bc, ce, db, ed , що утворить внутрішньо стійку множина для графа G^2 ; виявляється, $\alpha(G^2) = 5$. У загальній кількості знайдених n – буквених слів у коді виражається числом внутрішньої стійкості графа – добутку $\alpha(G^n)$.

Число внутрішньої стійкості

Цей приклад разом з теоремою 5 приводить до наступного визначення.

- ***Визначення.*** *Ємністю* графа G називається число $\theta(G) = \sup\limits_{n} \sqrt[n]{\alpha(G^n)}$. Тому що $\alpha(G^n) \leq |X^n| = |X|^n$, те $\theta(G) \leq |X|$.

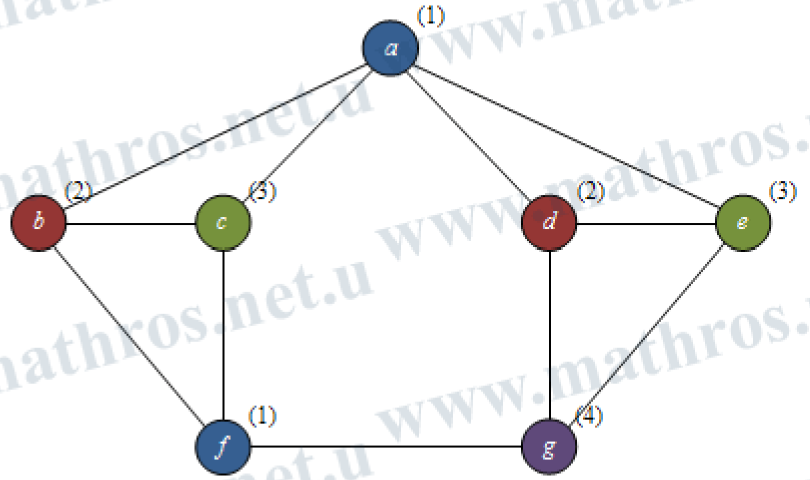
Наведена нижче теорема безпосередньо випливає з теореми 5. Вона дозволяє обчислити ємність для дуже широкого класу графів.

Теорема 6. Якщо для графа $G = G(X, \Gamma)$ існує зберігаюче відображення σ таке, що множина $\sigma(X)$ внутрішньо стійка, то для такого графа $\theta(G) = \alpha(G)$.

Переборний алгоритм для розфарбування вершин графа

Розфарбуванням вершин графа називається процес призначення певного кольору кожній з його вершин. Зазвичай кольори – це числа $1, 2, 3, \ldots, k$. Тоді, розфарбування є функцією, визначеною на множині вершин графа, яка приймає значення з множини $\{1, 2, 3, \ldots, k\}$.

Розмальовку можна також розглядати як розбиття множини вершин графа на підмножини, кожна з яких являється множиною вершин певного кольору. Відмітимо, що такі підмножини називаються кольоровими класами. Розфарбування називається правильним, якщо кожен кольоровий клас є незалежною множиною. Інакше кажучи, в правильному розфарбуванні будь-які дві суміжні вершини повинні мати різні кольори. Задача про розфарбовування полягає в знаходженні правильної розмальовки графа Gв найменше число кольорів. Це число називається **хроматичним числом графа** і позначається $\chi(G)$.



Розфарбування вершин графа найменшим набором кольорів

У правильному розфарбуванні повного графа G, кожна з його n вершин повинна бути зафарбована у свій колір, тому **хроматичне числа графа** такого типу дорівнює кількості його вершин, тобто $\chi(G) = n$. Якщо в графі існує повний підграф з k вершинами, то для розмальовки цього підграфа необхідно k кольорів. Звідси випливає, що для будь-якого графа виконується нерівність $\chi(G) \geq \omega(G)$, де $\omega(G)$ – клікове число графа G.

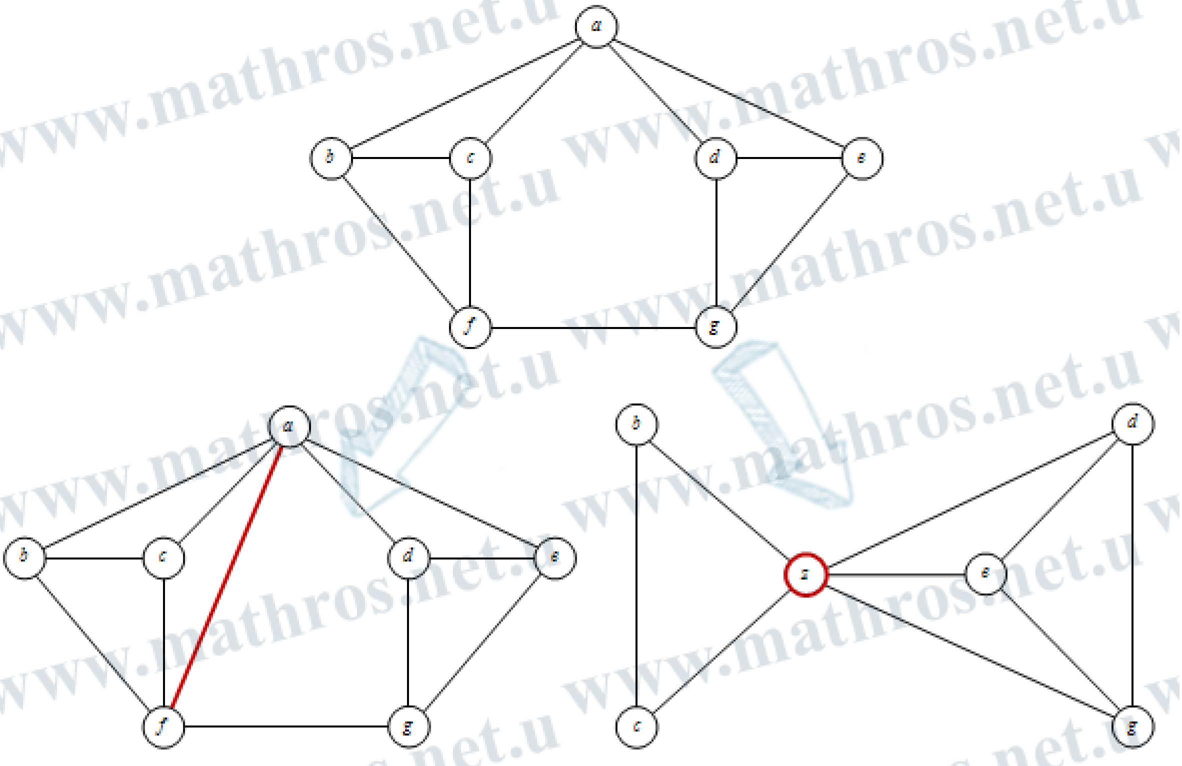
Відмітимо, що **хроматичне число графа** може бути і строго більше за його клікове число. Наприклад, на рисунку, що міститься вище, зображений граф, вершини якого розфарбовані в чотири кольори (номер кольору кожної вершини показано в дужках). Неважко перевірити, що трьох кольорів для правильного розфарбування цього графа недостатньо. Отже, його **хроматичної число** дорівнює чотири. Очевидно також, що клікове число цього графа дорівнює три.

Також хочеться зазначити, що, $\chi(G) = 1$ тоді і тільки тоді, коли G – порожній граф. Неважко охарактеризувати і графи з **хроматичним числом** рівним два (точніше, не більше два). За визначенням, це такі графи, множину вершин яких можна розбити на дві незалежних підмножини. Виходячи з того, що дане визначення збігається з визначенням **дводольного графа**, приходимо до висновку, що дводольні графи також називають біхроматичними. Тоді, скориставшись теоремою, яка вказує на критерій дводольності графа, запишемо критерій його біхроматичності. Отже, граф називається біхроматичним тоді і тільки тоді, коли всі його прості цикли мають парну довжину.

Для графів з **хроматичним числом** три та більше, такого простого опису невідомо. Невідомо й простих алгоритмів, які б дозволили перевірити, чи можна даний граф розфарбувати в три кольори. Проте, алгоритми для подібних задач розроблялися і продовжують розроблятися і в деяких випадках вони можуть бути корисні. Всі ці алгоритми в тій чи іншій формі здійснюють перебір варіантів (число яких може бути дуже великим).

Далі, розглянемо один із способів такого перебору для задачі про **розфарбовування графа**, який базується на ідеї, що задача такого типу, для заданого графа, зводиться до тієї ж задачі для двох інших графів. Тобто в результаті виконання даного алгоритму, виникає дерево варіантів, обхід якого дозволяє знайти рішення задачі. Розглянемо даний алгоритм більш детально.

Отже, в заданому графі G, виберемо дві не суміжні між собою вершини, наприклад a та f. На наступному кроці, побудуємо два нових графа: G_1, що виходить додаванням ребра (a, f) до графа G, і G_2, що виходить з G в результаті злиття вершин a та f (операція злиття полягає у видаленні вершин a та f і додаванні нової вершини, наприклад z, та ребер, що з’єднують її з кожною вершиною, з якою була суміжна хоча б одна з видалених вершин a і f).

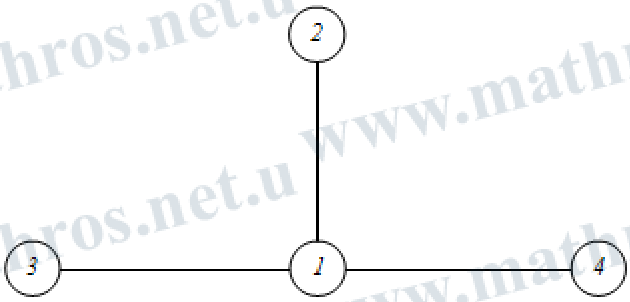


Використання алгоритму методу перебору для розв’язку задачі про розфарбування графа

Відмітимо, що якщо в правильному **розфарбуванні графа** G вершини a і f мають різні кольори, то вона буде правильною і для графа G_1. Якщо ж кольори вершин a та f у розфарбуванні графа G однакові, то граф G_2 можна розфарбувати в ту саму кількість кольорів: нова вершина фарбується в той колір, в який пофарбовані вершини a та f, а всі інші вершини зберігають ті кольори, які вони мали в графі G. Справедливе і обернене твердження, тобто розфарбування кожного з графів G_1 та G_2, очевидно, дає розмальовку графа G в ту саму кількість кольорів. Тому, $\chi(G) = \min(\chi(G_1), \chi(G_2))$, що дає можливість рекурсивного знаходження розмальовки графа в мінімальне число кольорів. Зауважимо, що граф G_1 має стільки ж вершин, скільки вихідний граф, але у нього більше ребер. Тому рекурсія в кінцевому рахунку призводить до повних графів, для яких задача про розфарбовування вирішується тривіально.

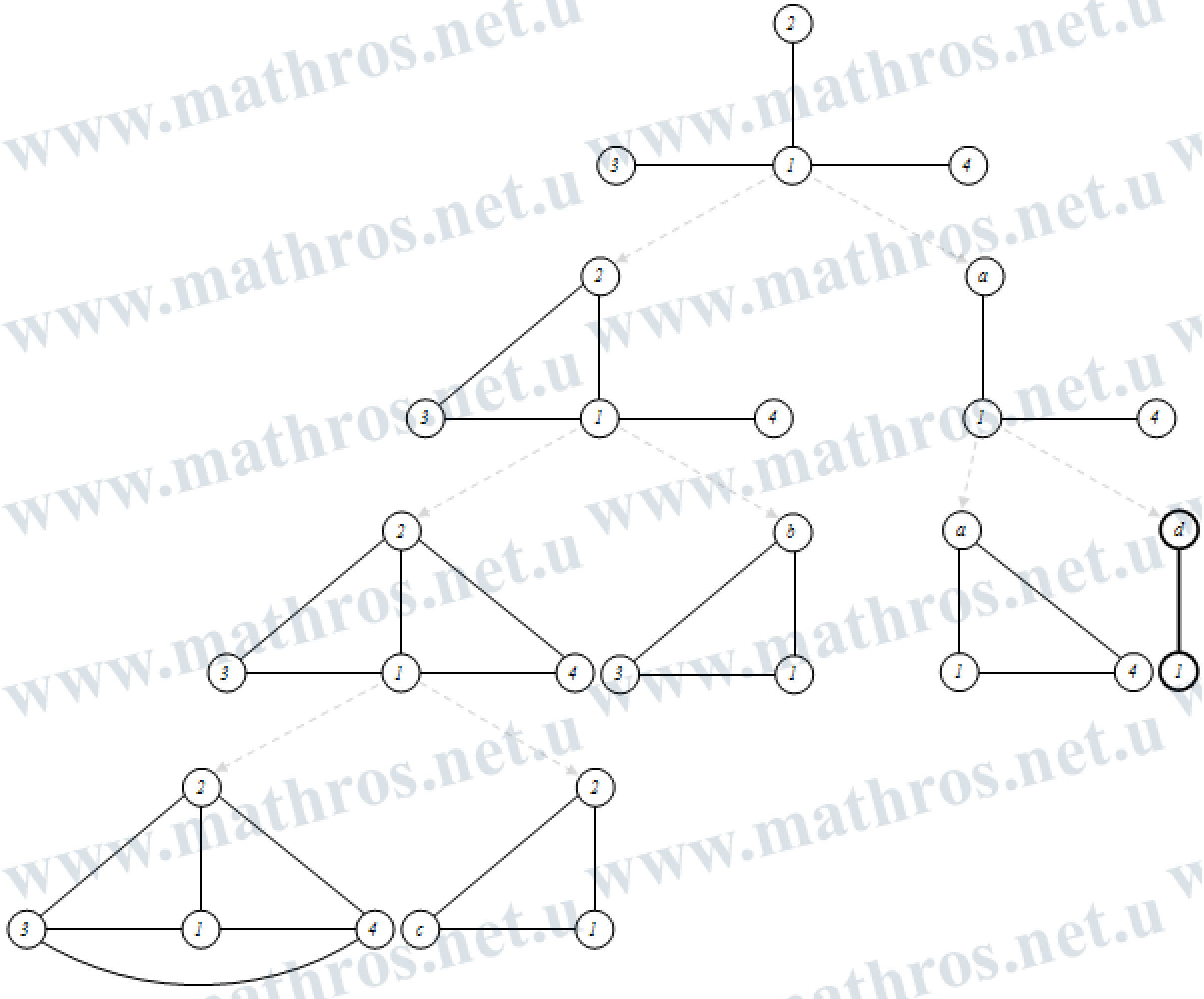
Розфарбування вершин графа використовуючи переборний алгоритм – приклад:

Знайти розв’язок задачі про **розфарбування вершин неорієнтованого графа** зображеного на наступному малюнку.



Неорієнтований граф задачі

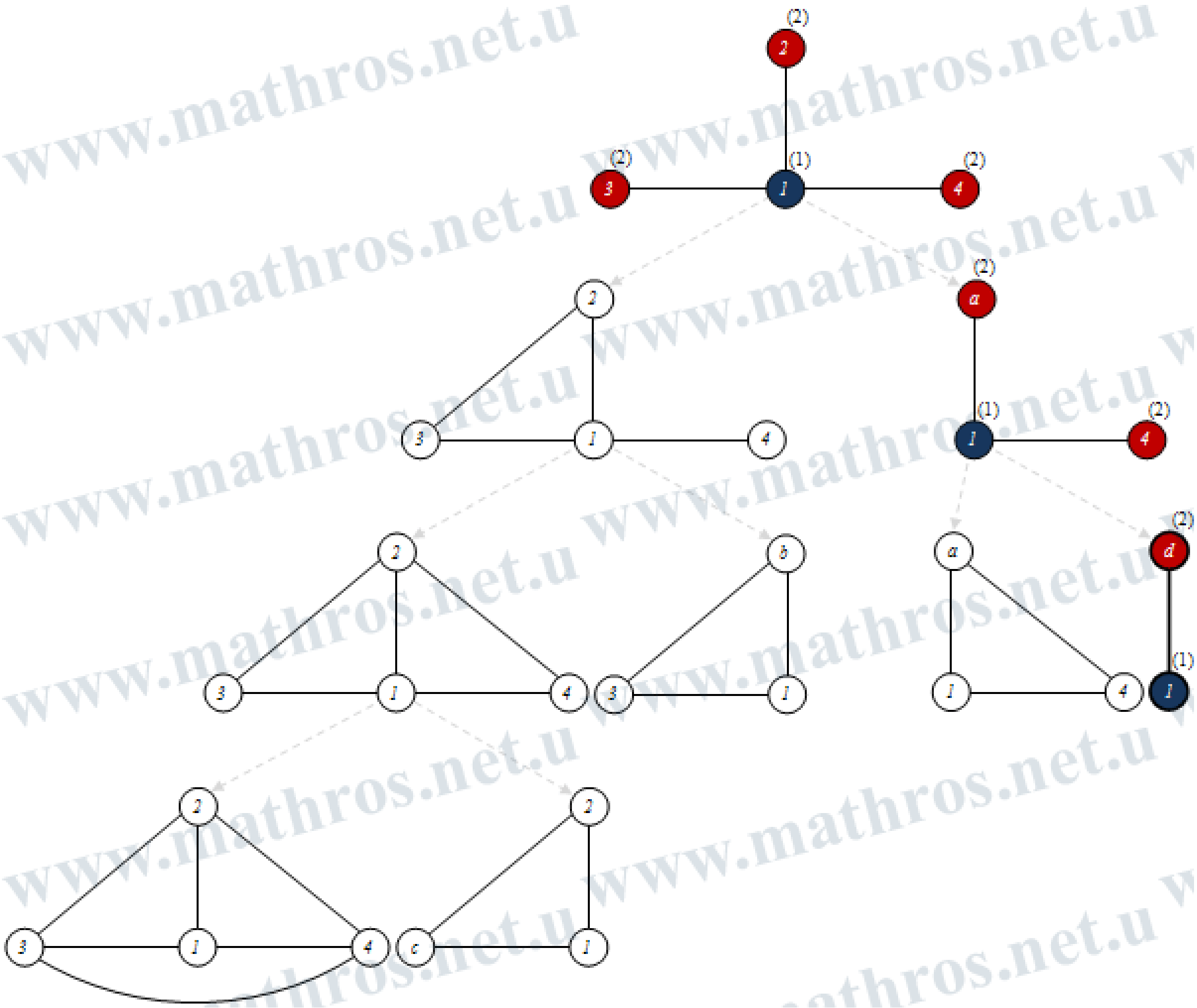
Для цього, скориставшись розглянутим вище алгоритмом, побудуємо дерево варіантів, обхід якого дозволить знайти шуканий розв’язок.



Дерево варіантів

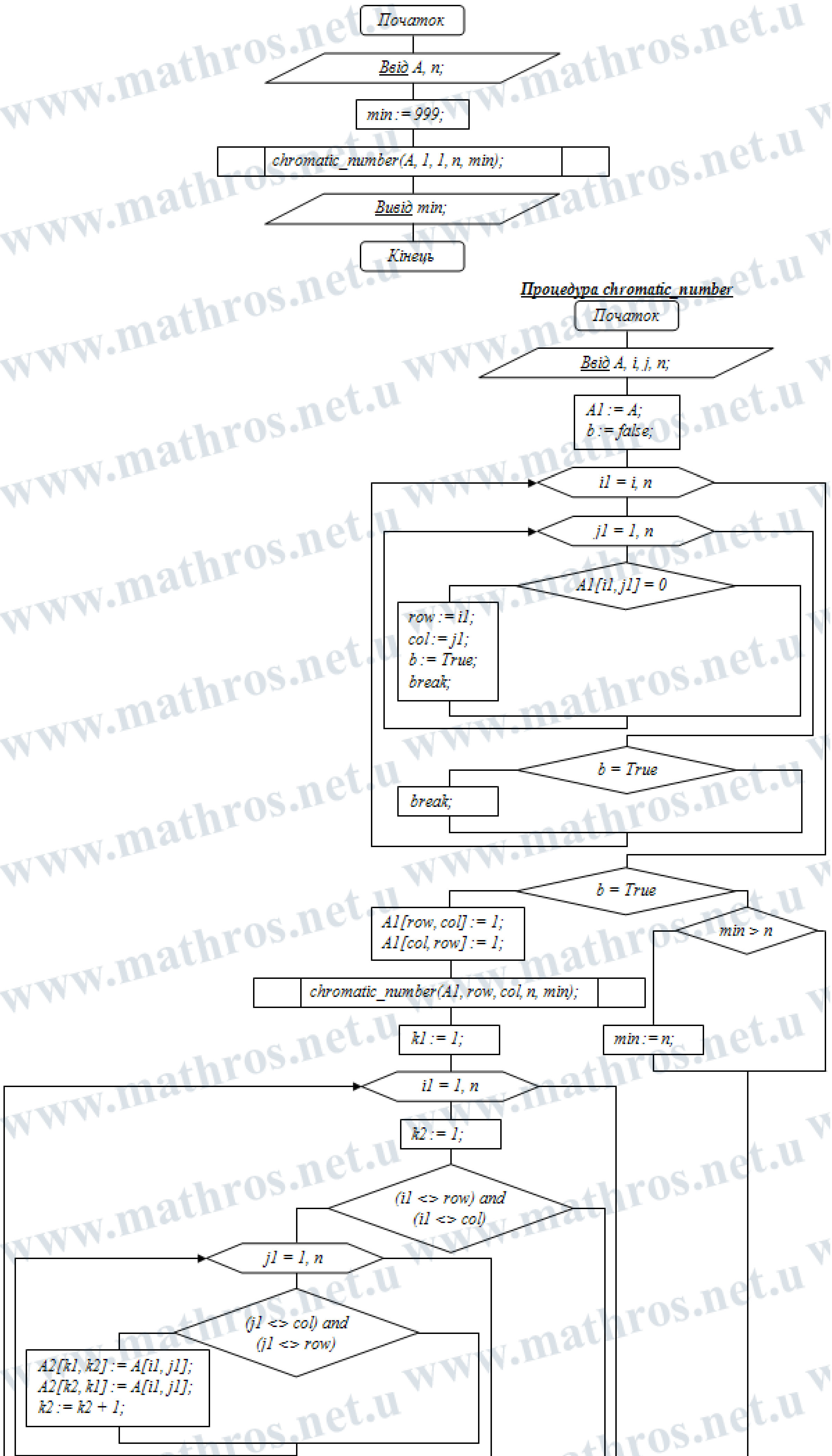
Переглянувши отримане дерево бачимо, що серед його зовнішніх вузлів міститься граф, для якого **хроматичне число** являється мінімальним і дорівнює два (на рисунку даний граф виділений жирними лініями). Звідси, приходимо до висновку, що для того, щоб здійснити правильне розфарбування заданого графа, знадобиться лише два кольори. Покажемо, яким чином це реалізується.

Отже, на першому кроці, розфарбуємо кожну вершину графа з мінімальним **хроматичним числом** в свій колір. Наприклад, вершину *«d»* в червоний колір, і вершину *«1»* – в синій. Після цього, виходячи з того, що вершина *«d»* була отримана в результаті злиття двох вершин *«4»* та *«a»* фарбуємо їх також в червоний колір. І на останньому кроці, з аналогічних міркувань, фарбуємо вершини *«2»* та *«3»* також в червоний колір. В результаті отримаємо:


(1)
(2)
a
b
c
d

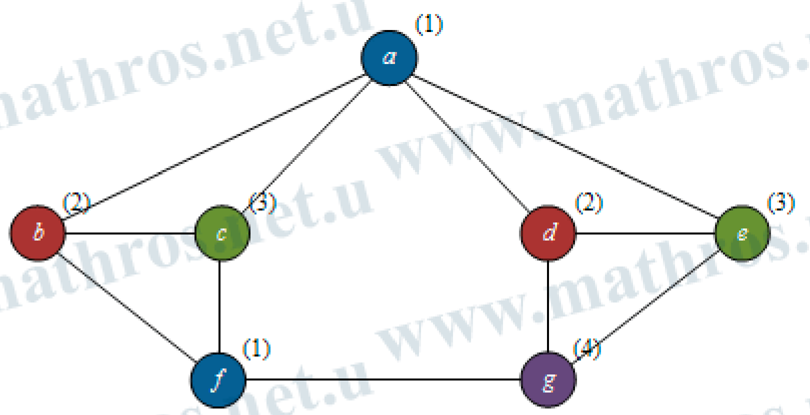
Правильне розфарбування вершин неорієнтованого графа

Блок-схема алгоритму знаходження числа кольорів необхідного для реалізації правильного розфарбування графа


Початок
Ввід A, n;
min := 999;
chromatic_number(A, 1, 1, n, min);
Вивід min;
Кінець
Процедура chromatic_number
Початок
Ввід A, i, j, n;
A1 := A;
b := false;
i1 = i, n
j1 = 1, n
A1[i1, j1] = 0
row := i1;
col := j1;
b := True;
break;
b = True
break;
b = True
min > n
A1[row, col] := 1;
A1[col, row] := 1;
chromatic_number(A1, row, col, n, min);
min := n;
k1 := 1;
i1 = 1, n
k2 := 1;
(i1 <> row) and
(i1 <> col)
j1 = 1, n
(j1 <> col) and
(j1 <> row)
A2[k1, k2] := A[i1, j1];
A2[k2, k1] := A[i1, j1];
k2 := k2 + 1;

Алгоритм послідовної розмальовки графа

Розглядаючи [задачу знаходження хроматичного числа графа](#) (мінімальна кількість кольорів, в які можна розфарбувати його вершини) нами зазначалось, що єдиним підходом для знаходження оптимального рішення задач такого типу є перебір варіантів. До прикладу в параграфі, що міститься за посиланням вище, нами було розглянуто переборний алгоритм, результатом виконання якого є дерево варіантів, обхід якого дозволяє знайти рішення задачі. Але, як видно з прикладу, навіть для графів з мінімальним числом вершин, отримане дерево може бути достатньо громіздким, не кажучи про графи кількість верин яких перевищує число десять і більше. Таким чином, перебрний метод при знаходженні оптимального розв'язку задачі на **розфарбування вершин графа** вимагає великих обчислювальних затрат і, як правило, застосовується для графів, з достатньо малою кількістю вершин.



Степеня вершин графа *G*	4	3	3	3	3	3	3
Вершини графа *G*	*a*	*b*	*c*	*d*	*e*	*f*	*g*
Вершини графа *G* по незростанню степенів	*a*	*b*	*c*	*d*	*e*	*f*	*g*
Ітерація №1	1					1	
Ітерація №2	1	2		2		1	
Ітерація №3	1	2	3	2	3	1	
Ітерація №4	1	2	3	2	3	1	4

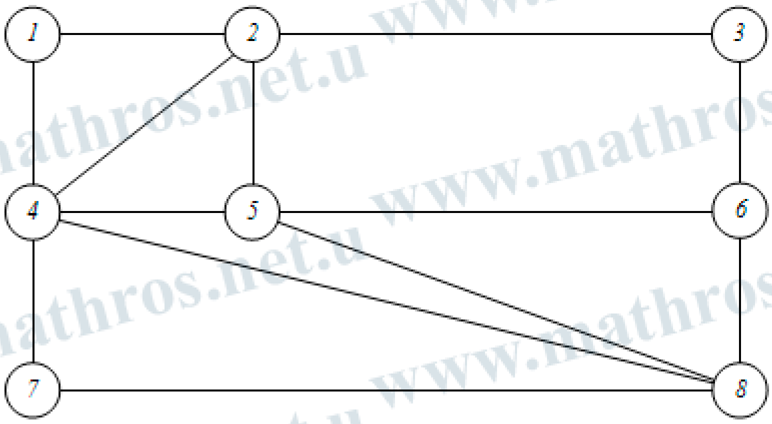
Використання алгоритму послідовної розмальовки для розв'язку задачі про розфарбування графа

Однак, якщо спростити розглядувану задачу і відмовитися від вимоги мінімальної кількості кольорів **розмальовки графа**, то можна побудувати алгоритми, які б працювали значно швидше, ніж алгоритм перебору варіантів. Зазначимо, що алгоритми, які швидко знаходять підходяще, але не оптимальне рішення, називаються евристичними. Прикладом раціонального евристичного алгоритму може слугувати **алгоритм послідовної розмальовки графа**, що базується на ідеї впорядкування вершин за будь-яким правилом (найчастіше зустрічаються приклади, де використовують степеня вершин) і послідовному присвоєнню їм кольорів, в які не були пофарбовані суміжні з ними вершини. Розглянемо даний алгоритм більш детально.

Отже, згідно з **алгоритмом послідовної розмальовки**, на першому кроці, вершини графа G розташовуються в порядку незростання їх степенів. Перша вершина фарбується в колір *«1»*. Після цього, список вершин проглядається зліва на право і в колір *«1»* фарбується всяка вершина, яка не є суміжною з іншою, вже пофарбованою в цей колір вершиною. На наступному кроці, повертаємося до першої не пофарбованої у списку вершини, фарбуємо її в колір *«2»* і знову переглядаємо список вершин зліва на право, фарбуючи в колір *«2»* будь-яку не пофарбовану вершину, яка не поєднана ребром з іншою, вже пофарбованої в колір *«2»* вершиною. Аналогічно діємо з кольорами *«3»*, *«4»* і так далі, і робимо це до тих пір поки не будуть пофарбовані всі вершини графа. Зазначимо, що число використаних кольорів приймають в якості наближеного значенням для **хроматичного числа графа**.

Алгоритм послідовної розмальовки графа – приклад:

Використовуючи послідовний алгоритм знайдемо наближений розв'язок задачі, яка полягає у знаходженні мінімального числа кольорів, в які можна **розфарбувати вершини графа** зображеного на наступному рисунку.

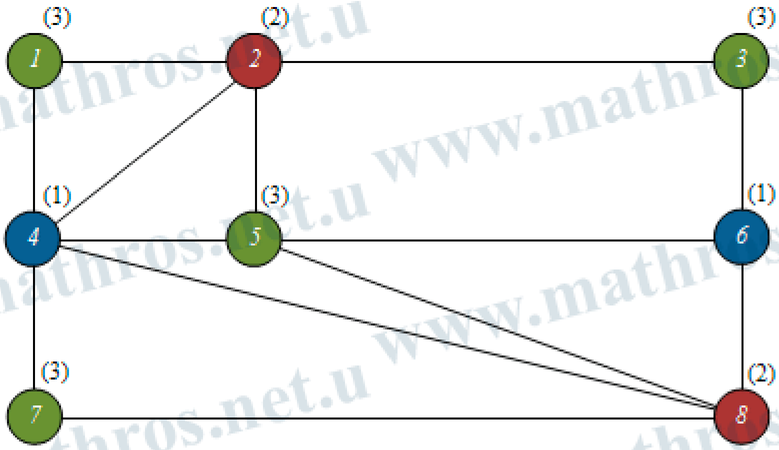


Неорієнтований граф задачі

Для цього, на першому кроці, множину вершини заданого графа запишемо в порядку незростання їх степенів. В результаті отримаємо: $\{4, 2, 5, 8, 6, 1, 3, 7\}$. Після цього, першу виршину з отриманої послідовності, а саме вершину номер чотири, фарбуємо, наприклад, в синій колір. Далі, список вершин проглядається зліва на право і в даний колір фарбується кожна вершина, яка не є суміжною з іншою, вже пофарбованою в цей колір вершиною. В нашому випадку, після виконання першої ітерації, в синій колір будуть пофарбовані вершини *«4»* та *«6»*.

Перейшовши до ітерації номер два, вибираємо першу не пофарбовану у списку вершину (в нашому випадку це вершина *«2»*), фарбуємо її в червоний колір і знову-таки переглядаємо список вершин зліва на право, фарбуючи в червоний колір будь-яку не пофарбовану вершину, яка не поєднана ребром з іншою, вже пофарбованої в даний колір вершиною. Зазначимо, що після виконання другої ітерації в червоний колір будуть пофарбовані вершини *«2»* та *«8»*.

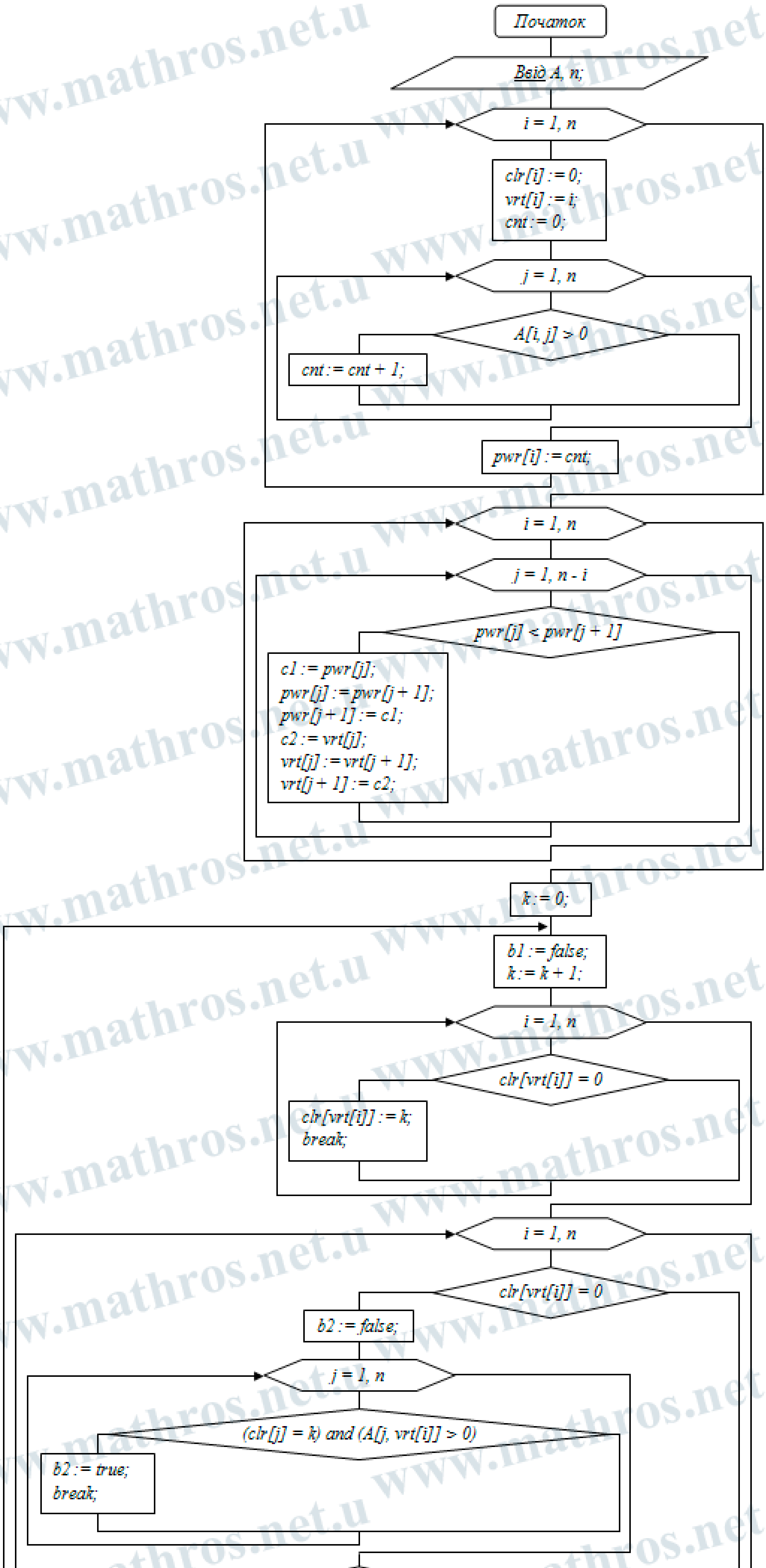
Прдовжуючи ітераційний процес далі, після виконання ітерації номер три, **алгоритм послідовної розмальовки** для заданого графа можна завершувати. Кожній з його вершин призначений певний колір (вершини *«1»*, *«3»*, *«5»* та *«7»* пофарбовані в зелений колір).



Степеня вершин графа *G*	2	4	2	5	4	3	2	4
Вершини графа *G*	1	2	3	4	5	6	7	8
Вершини графа *G* по незростанню степенів	4	2	5	8	6	1	3	7
Ітерація №1	1				1			
Ітерація №2	1	2		2	1			
Ітерація №3	1	2	3	2	1	3	3	3

Розфарбування вершин графа найменшим набором кольорів

Блок-схема алгоритму послідовної розмальовки графа


Початок
Ввід A, n;
i = 1, n
clr[i] := 0;
vrt[i] := i;
cnt := 0;
j = 1, n
A[i, j] > 0
cnt := cnt + 1;
pwr[i] := cnt;
i = 1, n
j = 1, n - i
pwr[j] < pwr[j + 1]
c1 := pwr[j];
pwr[j] := pwr[j + 1];
pwr[j + 1] := c1;
c2 := vrt[j];
vrt[j] := vrt[j + 1];
vrt[j + 1] := c2;
k := 0;
b1 := false;
k := k + 1;
i = 1, n
clr[vrt[i]] = 0
clr[vrt[i]] := k;
break;
i = 1, n
clr[vrt[i]] = 0
b2 := false;
j = 1, n
(clr[j] = k) and (A[j, vrt[i]] > 0)
b2 := true;
break;