





























**Переборний алгоритм для розфарбування вершин графа**

**Розфарбуванням вершин графа** називається процес призначення певного кольору кожній з його вершин. Зазвичай кольори – це числа . Тоді, розфарбування є функцією, визначеною на множині вершин графа, яка приймає значення з множини .

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 в найменше число кольорів. Це число називається **хроматичним числом графа** і позначається .



**Розфарбування вершин графа найменшим набором кольорів**

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

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, тоді і тільки тоді, коли  – 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**](https://www.mathros.net.ua/perevirka-grafa-na-dvodolnist.html), 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 , виберемо дві не суміжні між собою вершини, наприклад та . На наступному кроці, побудуємо два нових графа: , що виходить додаванням ребра до графа , і , що виходить з  в результаті злиття вершин  та  (операція злиття полягає у видаленні вершин  та  і додаванні нової вершини, наприклад , та ребер, що з’єднують її з кожною вершиною, з якою була суміжна хоча б одна з видалених вершин  і ).



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

Відмітимо, що якщо в правильному **розфарбуванні графа** вершини  і  мають різні кольори, то вона буде правильною і для графа . Якщо ж кольори вершин  та  у розфарбуванні графа  однакові, то граф  можна розфарбувати в ту саму кількість кольорів: нова вершина фарбується в той колір, в який пофарбовані вершини  та , а всі інші вершини зберігають ті кольори, які вони мали в графі . Справедливе і обернене твердження, тобто розфарбування кожного з графів  та , очевидно, дає розмальовку графа  в ту саму кількість кольорів. Тому, , що дає можливість рекурсивного знаходження розмальовки графа в мінімальне число кольорів. Зауважимо, що граф  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:



**Правильне розфарбування вершин неорієнтованого графа**

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



**Алгоритм послідовної розмальовки графа**

Розглядаючи [**задачу знаходження хроматичного числа графа**](https://www.mathros.net.ua/perebornyj-algorytm-dlja-rozfarbuvannja-vershyn-grafa.html) (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.



**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  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: . 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).



**Розфарбування вершин графа найменшим набором кольорів**

**Блок-схема алгоритму послідовної розмальовки графа**

