РОЗФАРБУВАННЯ ГРАФІВ

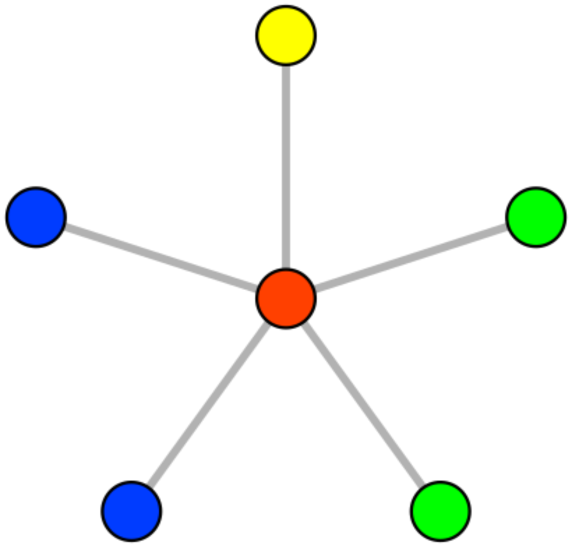
ЯРОШЕПТА БОГДАН

МЕТА

1. Систематизація основних означень та теорем про хроматичне число, хроматичний індекс та хроматичний поліном графа.
2. Розгляд жадібного алгоритму розфарбування графів, доведення його оптимальності, оцінка складності та недоліки.
3. Дослідження хроматичних поліномів.

ВСТУП

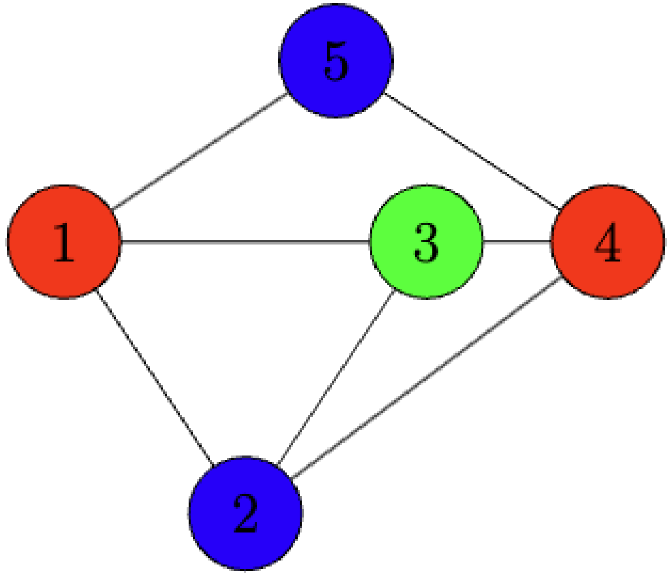
Розфарбування графів є важливою проблемою теорії графів, яка має широке застосування в різних галузях, включаючи телекомунікації, графічний дизайн та багато інших.



РОЗФАРБУВАННЯ ГРАФІВ [1]

Нехай $G = (V, E)$ - довільний граф, а $N_k = \{1,2,\dots,k\}$

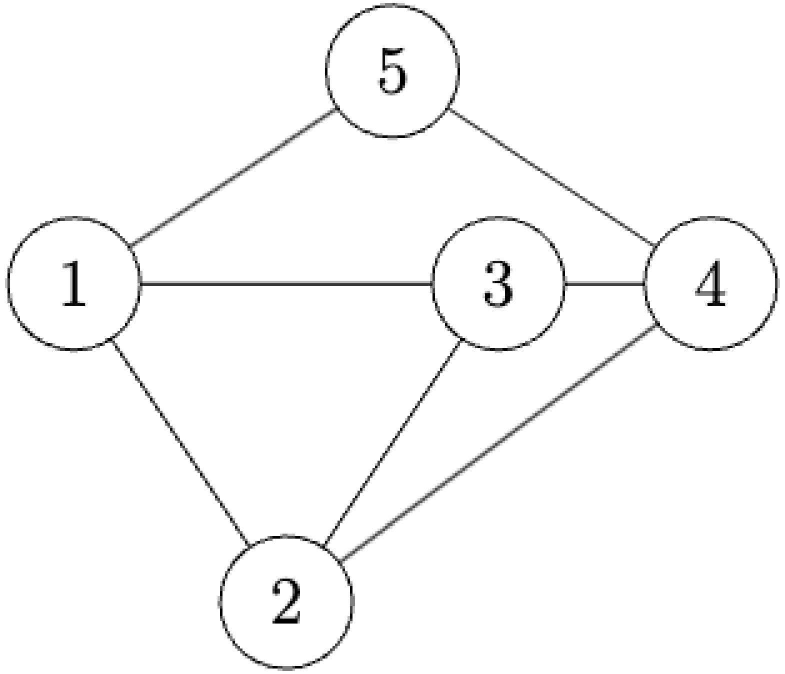
- Будь-яке відображення $f: V \rightarrow N_k$ яке кожній вершині $v \in V$ ставить у відповідність деяке натуральне число $f(v) \in N_k$ називають **кольором**, або **номером фарби**, вершини v.
- Розфарбування f графа G називають **правильним**, якщо для будь-яких його суміжних вершин v і w виконується $f(v) \neq f(w)$
- Мінімальне число k, для якого існує правильне розфарбування вершин графа G, називають **хроматичним числом** графа G і позначають $\chi(G)$
- **Хроматичний індекс** графа G, позначений як $\chi'(G)$, є найменшою кількістю кольорів, необхідних для розфарбування ребер графа таким чином, що жодні два суміжних ребра не мали однакового кольору. Тобто для довільної пари ребер $e_1 = (u, v)$ та $e_2 = (v, w)$, які мають спільну вершину v, маємо: якщо e_1 та e_2 розфарбовані кольорами c_1 та c_2 відповідно, то $c_1 \neq c_2$.



ЖАДІБНИЙ АЛГОРИТМ [2]

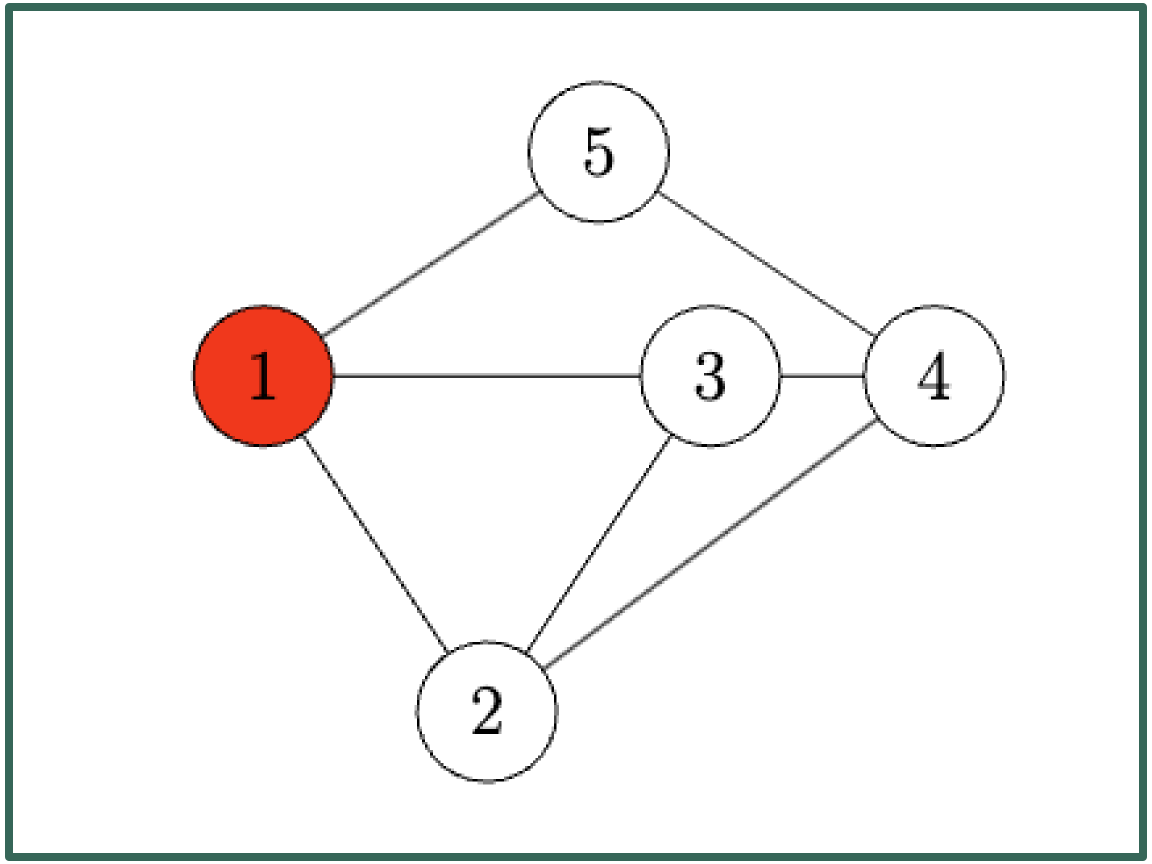
- Жадібний алгоритм розфарбування графів є простим і ефективним підходом до розв'язання задачі розфарбування графів.
- Він може бути швидким і простим у реалізації, але не завжди гарантує оптимальне розфарбування. В окремих випадках може знадобитися більше кольорів, ніж алгоритм використовує. Проте, цей алгоритм може слугувати як початкове рішення та база для подальших вдосконалень та складніших алгоритмів розфарбування графів.
- Для використання жадібного алгоритму, спочатку всі вершини графа отримують номери випадковим чином, після чого послідовно кожну вершину розфарбовують в колір з мінімальним номером, який відсутній у суміжних з нею вершин.
- Відповідно, маємо твердження:
  - Вершини будь-якого графу G можна пронумерувати так, що жадібний алгоритм використає рівно $\chi(G)$ кольорів

ЖАДІБНИЙ АЛГОРИТМ



- Розглянемо приклад використання жадібного алгоритму для розфарбування деякого графу.
- На ілюстрації зображено граф, який будемо розфарбовувати.
- Визначимо номери кольорів:
  - 1 – червоний
  - 2 – синій
  - 3 – зелений

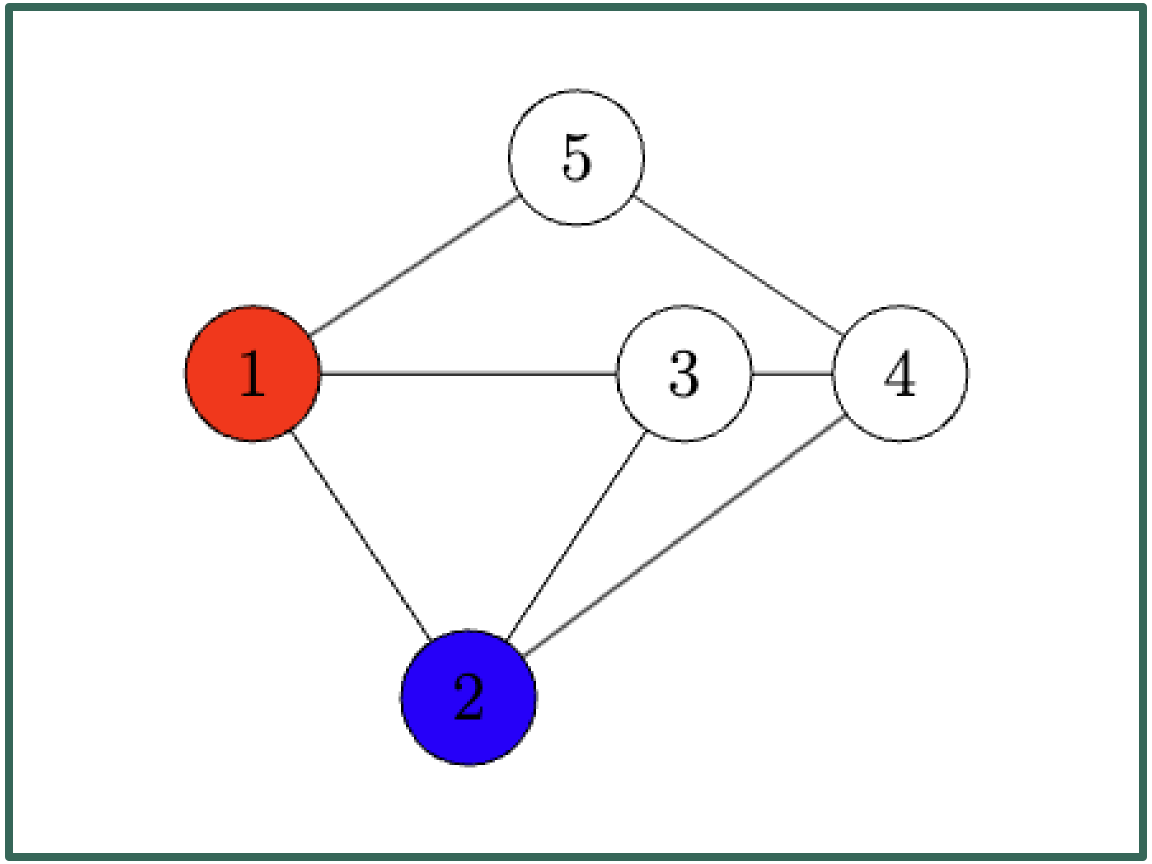
ЖАДІБНИЙ АЛГОРИТМ



Крок 1: Початкова вершина

- Починаємо з обраної початкової вершини, а саме вершина з номером 1, і призначимо їй найменший доступний колір (червоний).

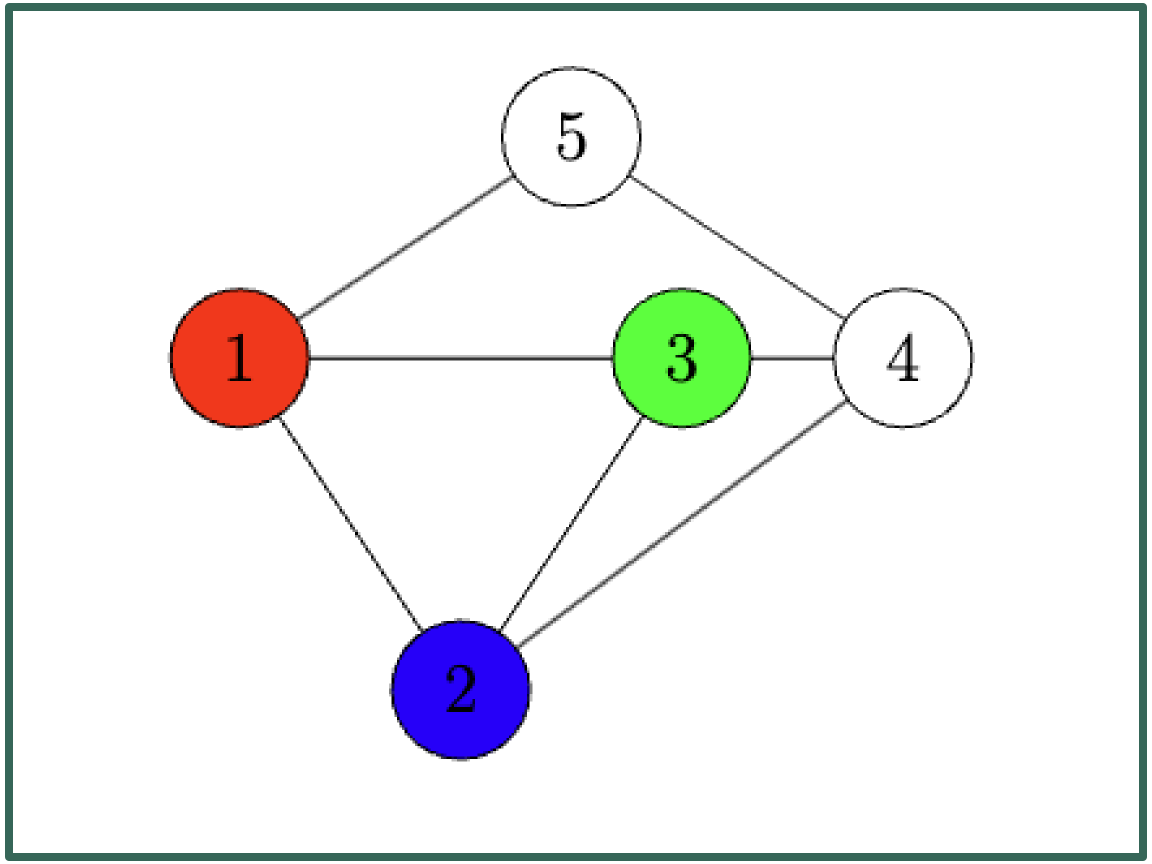
ЖАДІБНИЙ АЛГОРИТМ



Крок 2: Наступна вершина

- Далі ми продовжуємо розфарбовування графу.
- Переходимо до наступної вершини і перевіряємо кольори суміжних з нею вершин.
- Бачимо, що єдина розфарбована вершина, з якою вону суміжна має червоний колір (1),
- Отже, фарбуємо вершину 2 в найменший доступний колір, а саме синій (2)

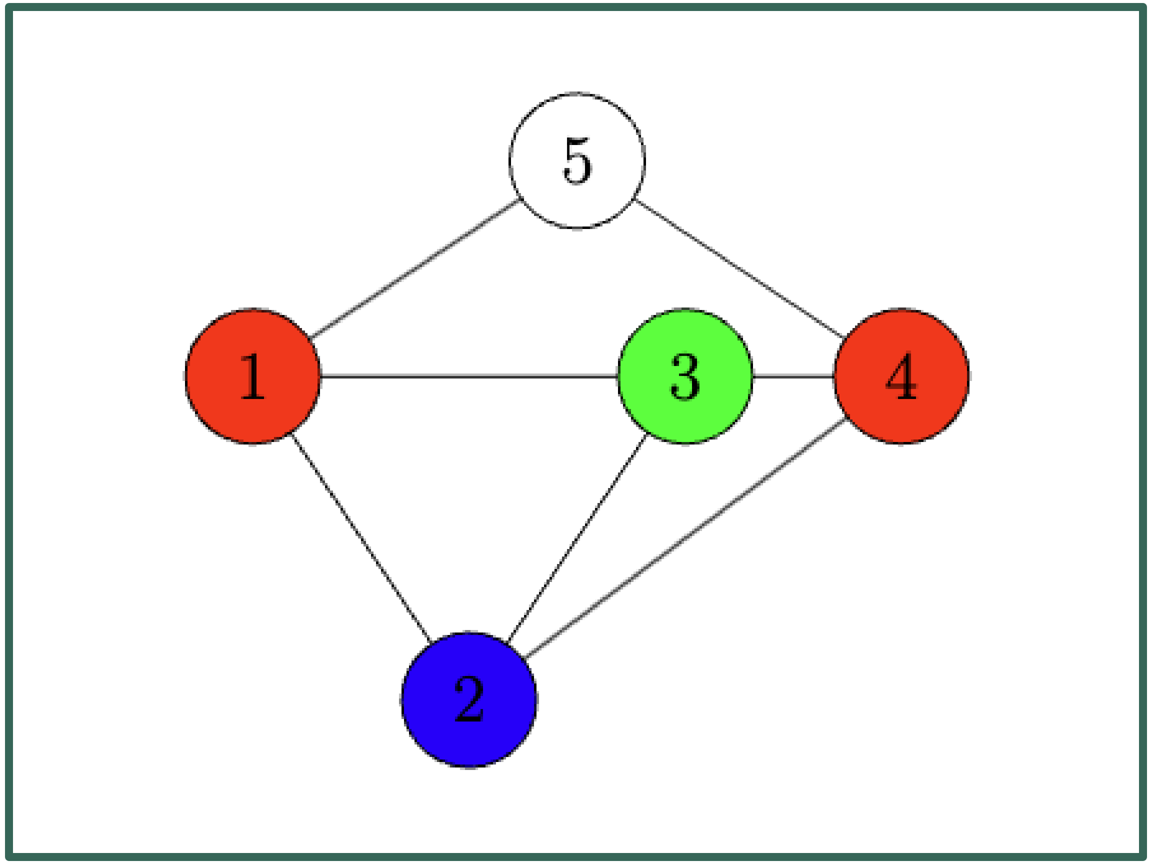
ЖАДІБНИЙ АЛГОРИТМ



Крок 3: Наступна вершина

- Переходимо до наступної вершини і знову перевіряємо кольори її сусідів.
- Бачимо, що суміжні з нею вершини вже пофарбовані в червоний(1) та синій(2) кольори.
- Отже, фарбуємо вершину 3 в зелений(3) колір

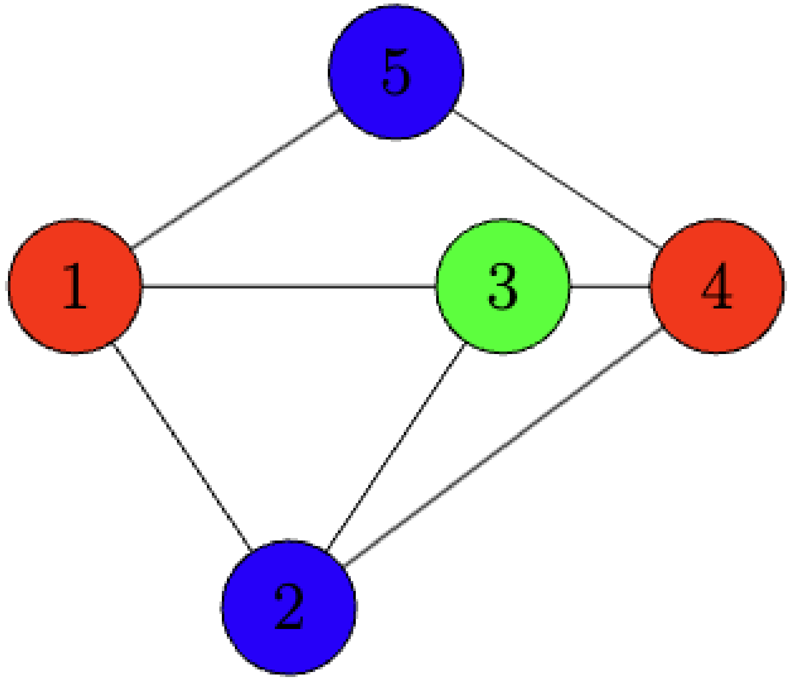
ЖАДІБНИЙ АЛГОРИТМ



Крок 4: Продовження розфарбовування

- Переходимо до вершини 4
- Аналогічно до попередніх кроків визначаємо найменший доступний колір для цієї вершини.
- Фарбуємо вершину 4 в червоний (1) колір

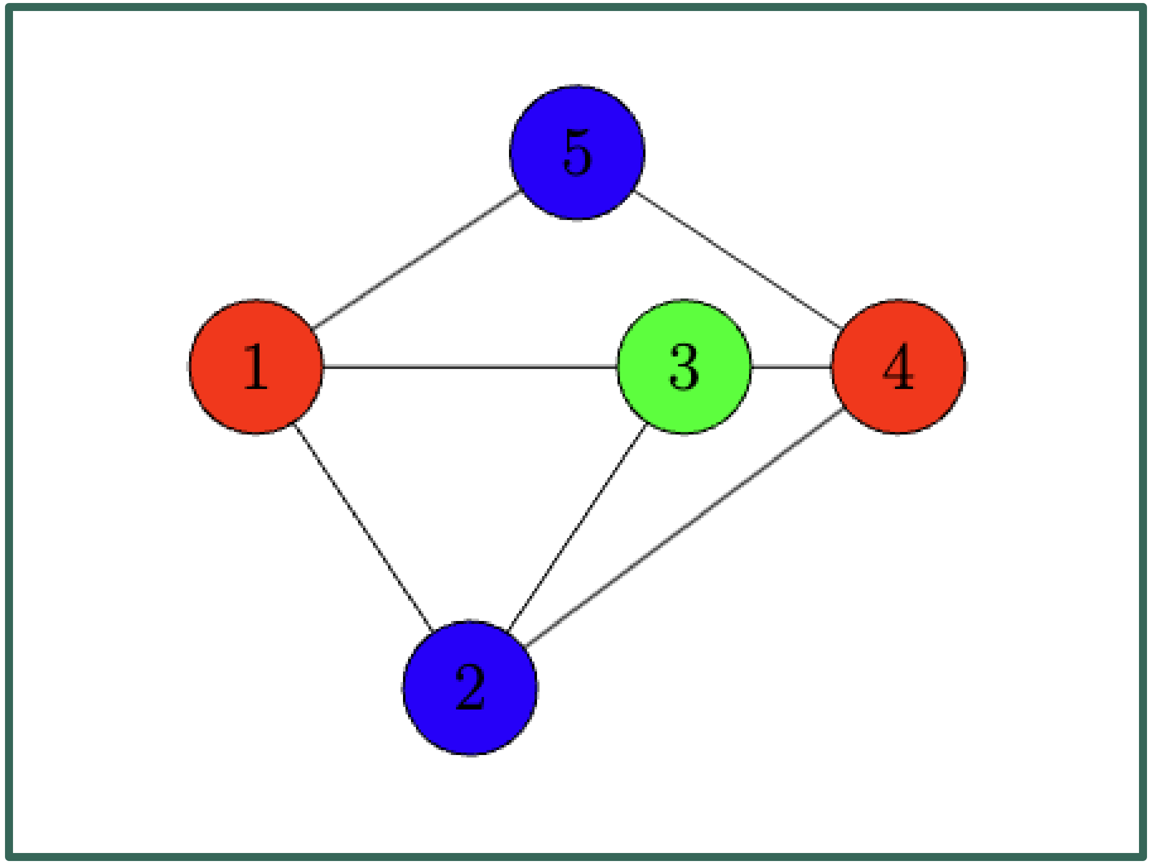
ЖАДІБНИЙ АЛГОРИТМ



Крок 5: Остання вершина

- Тепер розглянемо останню не розфарбовану вершину.
- Бачимо, що суміжні з нею вершини мають тільки червоний (1) колір.
- Отже, в нас є два варіанта розфарбування цієї вершини.
- Але за правилами жадібного алгоритму ми фарбуємо вершину 5 в найменший доступний колір, а саме в синій (2).

ЖАДІБНИЙ АЛГОРИТМ



Результат

- На цьому слайді ми бачимо результат розфарбовування графу за допомогою жадібного алгоритму.
- Кожна вершина отримала призначений колір, з урахуванням обмежень розфарбовування.
- Тому, маємо правильне розфарбування.

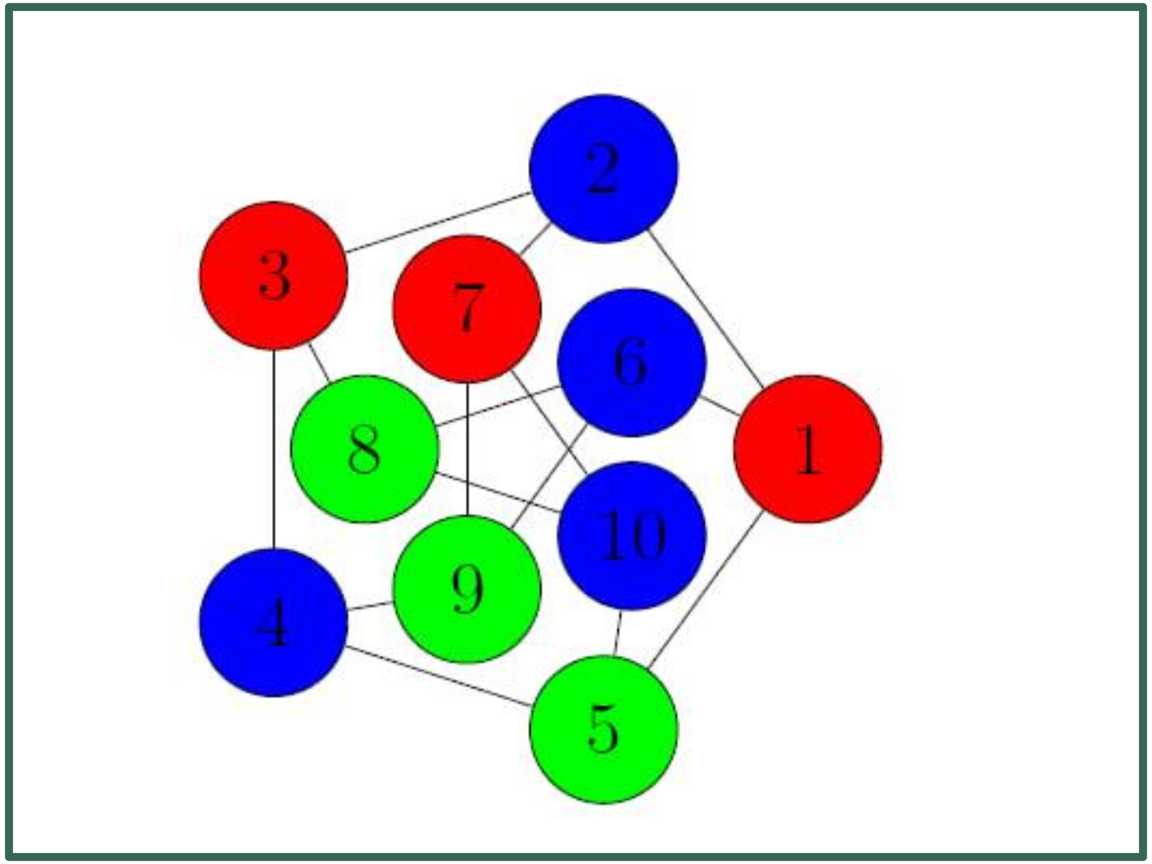
ЖАДІБНИЙ АЛГОРИТМ

Функція розфарбування

Напишемо функцію для жадібного розфарбування графа.

```
def greedy_coloring(graph):
    colors = {}
    for node in graph:
        available_colors =
         set(range(len(graph))) - {colors.get(neighbor)
         for neighbor in graph[node]}
        colors[node] = min(available_colors)
    return colors
```

ЖАДІБНИЙ АЛГОРИТМ



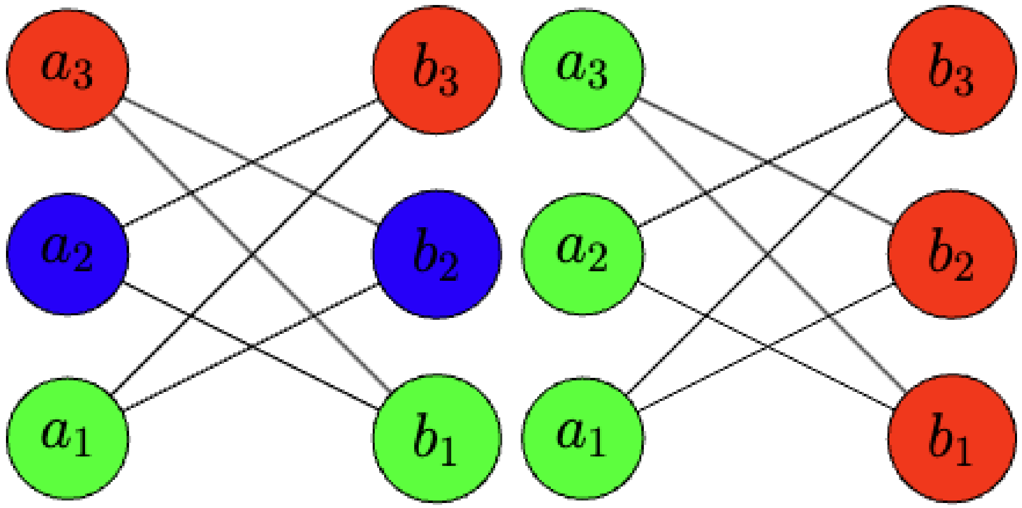
ЖАДІБНИЙ АЛГОРИТМ

Недоліки жадібного алгоритму розфарбування графів:

- Не гарантує оптимальне розфарбування: жадібний алгоритм може призводити до розфарбування графу, яке вимагає більшої кількості кольорів, ніж оптимальне розфарбування.
- Залежність від порядку вузлів: жадібний алгоритм може давати різні результати в залежності від порядку вершин, у якому вони розфарбовуються. Різний порядок може призводити до різної кількості використаних кольорів.
- Не ефективний для деяких типів графів: жадібний алгоритм може бути неефективним для графів зі складною структурою, таких як графи з великою кількістю циклів або графи зі складною геометричною конфігурацією.
- Залежність від вибору першої вершини: початковий вибір вершини для розфарбування може впливати на результат. Різні початкові вершини можуть призводити до різних розфарбувань.

ЖАДІБНИЙ АЛГОРИТМ

- На ілюстрації можемо побачити приклад двох розфарбувань одного двочасткового графа, які відрізняються тільки порядком вибору вершин.
- В першому (неоптимальному) випадку, ви починаємо з вершини a_1, розфабовуємо її в зелений колір та потім переходимо до вершини b_1, відповідно, розфарбовуючи її також в зелений колір і тд.
- В другому випадку ми так само починаємо з вершини a_1 та потім переходимо до вершини a_2 та a_3, що в результаті дає нам оптимальне розфарбування.



ТЕОРЕМА ВІЗІНГА ДЛЯ ХРОМАТИЧНОГО ЧИСЛА [3]

- Для будь-якого простого графа G, хроматичне число $\chi(G)$ графа G можна обмежити двома способами:
  - Нижня межа Візінга: $\chi(G) \geq \omega(G)$, де $\omega(G)$ - незалежне число графа G, що представляє максимальну кількість взаємно непов'язаних вершин (вершин, які не з'єднані ребрами).
  - Верхня межа Візінга: $\chi(G) \leq \Delta(G) + 1$, де $\Delta(G)$ - максимальна степінь вершини в графі G.
- Отже, хроматичне число графа G знаходиться між нижньою і верхньою межами Візінга.

НАСЛІДОК З ТЕОРЕМИ ВІЗІНГА[3]

- Для будь-якого простого графа G, хроматичний індекс $\chi'(G)$ графа G можна обмежити двома способами:
  - Нижня межа Візінга: $\chi'(G) \geq \Delta(G)$, де $\Delta(G)$ - максимальна степінь вершини в графі G.
  - Верхня межа Візінга: $\chi'(G) \leq \Delta(G) + 1$, де $\Delta(G)$ - максимальна степінь вершини в графі G.
- Отже, хроматичний індекс графа G знаходиться між нижньою і верхньою межами Візінга.

ТЕОРЕМА БРУКСА [3]

- Для будь-якого простого зв'язного графа G зі степенем кожної вершини не більшим за Δ (де Δ - максимальна ступінь вершини в графі G), хроматичне число $\chi(G)$ графа G не перевищує Δ, за винятком двох випадків: для повних графів і для непарних циклів довжиною більше 3, де хроматичне число буде дорівнювати $\Delta + 1$.

ХРОМАТИЧНИЙ ПОЛІНОМ [1]

- Хроматичний поліном $P(G,\lambda)$ графа G - це поліном, який визначається як кількість способів розфарбування вершин графа G за допомогою λ кольорів таким чином, щоб жодні дві суміжні вершини не мали однаковий колір.
- Формально, хроматичний поліном $P(G,\lambda)$ графа G визначається як:

  $P(G,\lambda) = \lambda(\lambda-1)(\lambda-2)\ldots(\lambda-n+1)$, де n - кількість вершин у графі G.
- Хроматичний поліном використовується для вивчення розфарбування графів та встановлення меж хроматичного числа, яке є найменшою кількістю кольорів, необхідних для розфарбування графа G. Властивості хроматичного поліному можуть надати важливу інформацію про структуру графа та його розфарбування.

ТЕОРЕМА ПРО РЕКУРЕНТНІ СПІВВІДНОШЕННЯ ДЛЯ ХРОМАТИЧНОГО ПОЛІНОМА [1]

- Рекурентний метод обчислення хроматичного многочлена базується на операції стягування ребра – для пари вершин v та w граф G/vw отримується шляхом стягування ребра vw.
- Хроматичний многочлен задовольняє наступному рекурентному співвідношенню: $P(G,k) = P(G - vw, k) - P(G\backslash vw, k)$, де v та w – суміжні вершини та $G - vw$ – граф з видаленим ребром vw.
- Еквівалентно $P(G,k) = P(G + vw) + P(G/vw, k)$, якщо v та w не суміжні та $G + vw$ – граф з доданим ребром vw

ТЕОРЕМА ПРО ВЛАСТИВОСТІ ХРОМАТИЧНОГО ПОЛІНОМА [1]

- Для будь-якого графа G функція $\chi_G(k)$ має наступні властивості:
  1. Функція $\chi_G(k)$ є многочленом з цілими коефіцієнтами;
  2. Степінь $\chi_G(k)$ дорівнює кількості вершин G;
  3. Старший коефіцієнт дорівнює 1;
  4. Коефіцієнт при доданку k^{n-1} від'ємний та за модулем рівний кількості ребер в графі G;
  5. Коефіцієнт при k^i дорівнює 0 тоді і тільки тоді, коли i менше за кількість компонент зв'язності графу G (зокрема, вільний член завжди дорівнює 0);

ТЕОРЕМА ВІТНІ [1]

- Нехай G – простий граф з n вершинами, m ребрами та k компонентами зв'язності.
- Тоді його хроматичний многочлен буде мати вигляд

$$f_G(x) = x^n - a_1 x^{n-1} + a_2 x^{n-2} - \cdots + (-1)^{n-k} a_{n-k} x^k,$$

- де $a_1, a_2, \dots, a_{n-k}$ - додатні цілі числа, при чому $a_1 = m$.

ЗНАХОДЖЕННЯ ХРОМАТИЧНОГО ПОЛІНОМУ ДЛЯ КЛАСІВ ГРАФІВ

- Нехай G – це граф триангуляції *n*-кутника, тоді його хроматичний многочлен:

$$f_G(x) = x(x-1)(x-2)^{n-2}$$

- Для графа-колеса на *n* вершинах:

$$f_{W_n}(x) = x(x-2)\left((x-2)^{n-2} + (-1)^{n-1}\right)$$

СПИСОК ЛІТЕРАТУРИ

1. F. Harary Graph Theory. Addison-Wesley, Reading, Mass., 1969.
2. U. S. R. Murty, Adrian Bondy Graph Theory: An Advanced Course Springer-Verlag London 2008
3. Reinhard Diestel. Graph Theory. Electronic Edition 2005 Springer-Verlag Heidelberg, New York 1997, 2000, 2005.
4. M. Aigner, G.M. Ziegler. Proofs from THE BOOK, Springer, 2003.

ДЯКУЮ ЗА УВАГУ

ЯРОШЕПТА БОГДАН, 2023 РІК