Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Дискретна математика

Лабораторна робота №4

«Розфарбовування графа, алгоритми розфарбування»

Виконала:

студентка групи ІО-15

Кушнерик Є.О.

Залікова книжка № 1508

Перевірив Пономаренко А. М.

Київ

2022 р.

**Зміст:**

1. Тема лабораторної роботи.
2. Мета лабораторної роботи.
3. Загальне завдання.
4. Короткі теоретичні відомості, що були використані при написанні лабораторної роботи.
5. Варіант виразу відповідно до індивідуального завдання. Спрощення виразу.
6. Аналіз результатів та висновки.

1.**Тема:** Розфарбовування графа, алгоритми розфарбування.

2. **Мета:** вивчення способів правильного розфарбовування графа..

3. **Загальне завдання**

1. Написати в окремому модулі функцію для формування несуперечливих бінарних відношень.

2. Написати в окремому модулі функції виконання логічних операцій над бінарними відношеннями.

3. Пояснити правило формування несуперечливих відношень відповідно до Вашого варіанту.

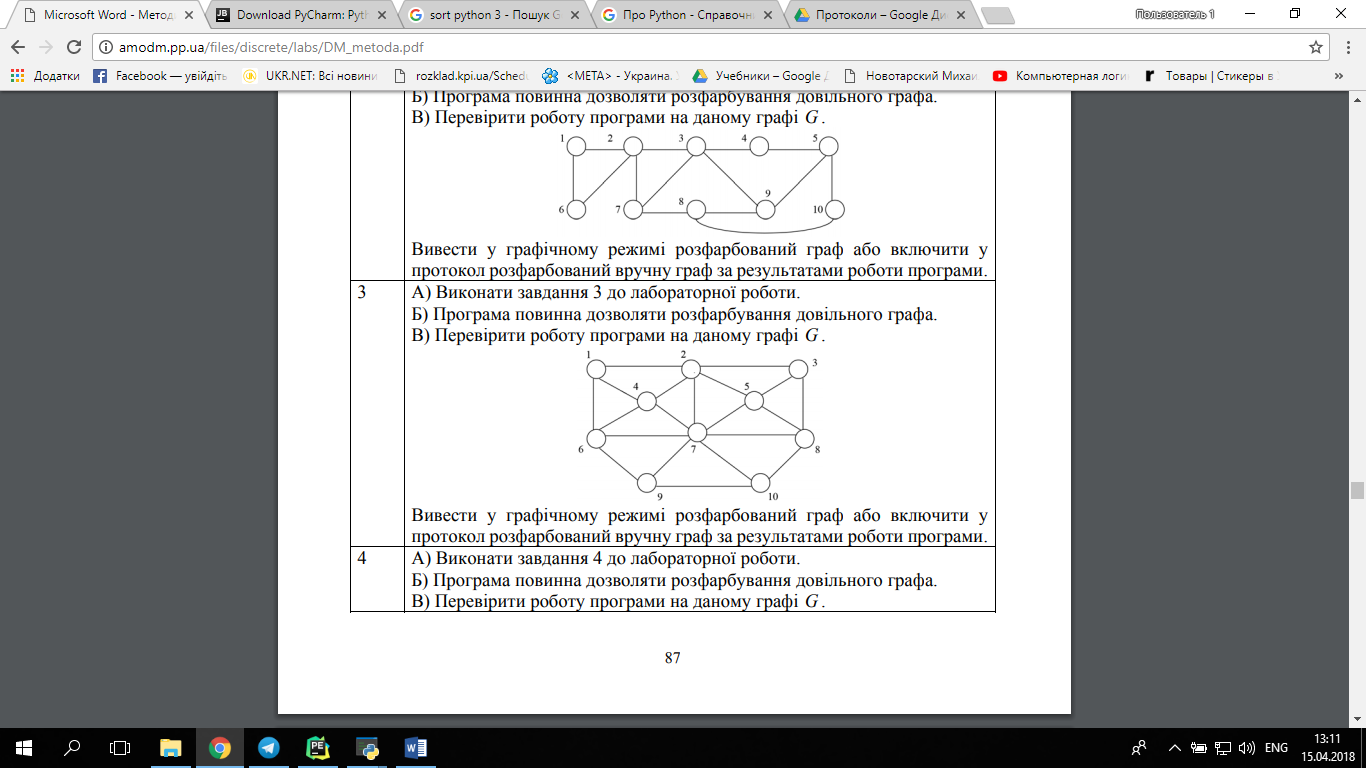
4. **Короткі теоретичні відомості:**

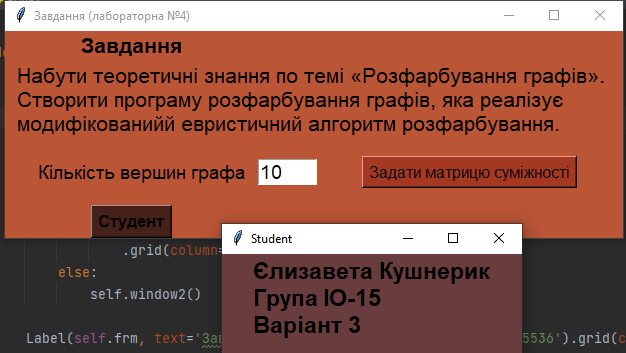
Модифікований евристичний алгоритм розфарбування:

Проста модифікація описаної вище евристичної процедури базується на переупорядкуванні нерозфарбованих вершин по незростанню їх відносних степенів. Дана модифікація полягає у тому, що якщо дві вершини мають однакові степені, то порядок таких вершин випадковий. Їх можна впорядкувати по двокрокових степенях. Двокроковий степінь визначимо як суму відносних степенів суміжних вершин.

5. **Варіант виразу відповідно до індивідуального завдання:**

Завдання варіанту:

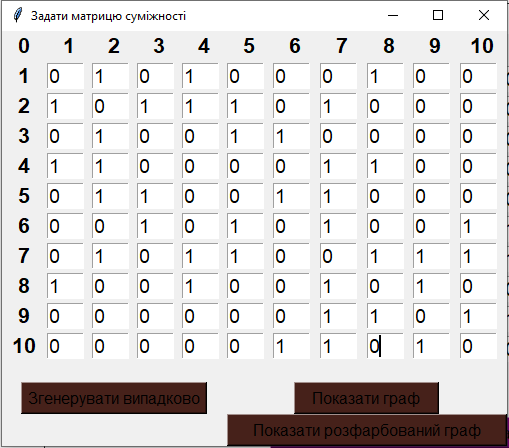




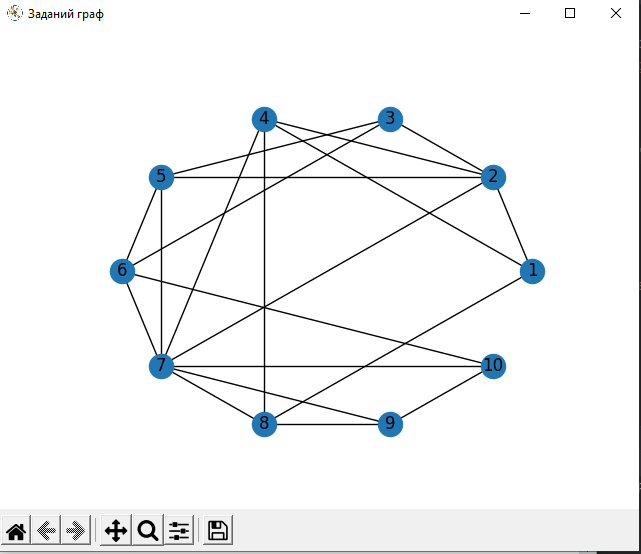
Алгоритм генерування відношень:

def GraphColoring(self, Edges):  
 result = {}  
 colors = ['#46211a', '#6e6702', '#c05805', '#db9501']  
  
 def SortEdges():  
 sort\_list = {}  
 for key in Edges:  
 sort\_list[key] = len(Edges[key])  
 sort2 = sorted(sort\_list.items(), key=lambda x: x[1], reverse=True)  
  
 return sort2  
  
 def colorize():  
 n = 0  
  
 def remove\_colour():  
 for colour in colors:  
 if colour not in stack\_colour: return colour  
 colour = colors[n + 1]  
 colors.append(colour)  
 return colour  
  
 for key in sort\_list:  
 stack\_colour = []  
 for key2 in Edges[key[0]]:  
 if key2 in result: stack\_colour.append(result[key2])  
 color = remove\_colour()  
 result[key[0]] = color  
 return result  
  
 sort\_list = SortEdges()  
 return colorize()

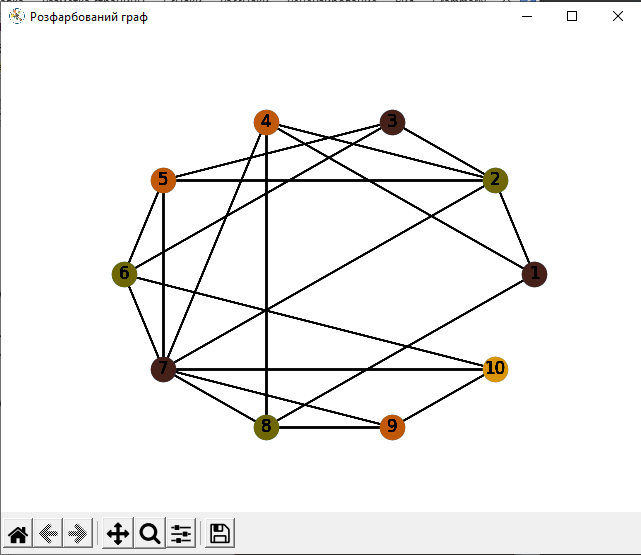
Матриця суміжності



Граф без змін



Розфарбований граф



6.**Висновок**

У ході виконання лабораторної роботи я ознайомилась з алгоритмами

розфарбування графів та реалізувала один із них на мові програмування Python.

7.**Контрольні запитання**

**1. Сформулюйте задачу розфарбовування графа.**

- Задача розфарбування графа полягає у тому, щоб кожній вершині

графа встановити у відповідність свій колір таким чином, щоб жодні

дві суміжні вершини не мали однакових кольорів, причому, щоб

загальна кількість кольорів була мінімально можливою.

- Задача розфарбування графа полягає у тому, щоб знайти хроматичне

число графа

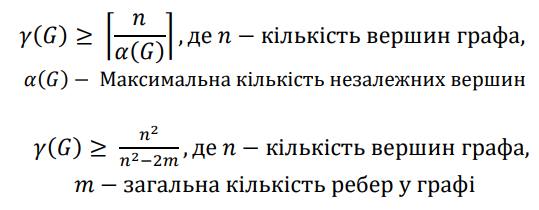
**2. Який граф називається r-хроматичним?**

Такий граф, який можна розфарбувати використовуючи r кольорів.

**3. Що називається хроматичним числом графа?**

Хроматичним числом графа називають найменш можливу кількість кольорів у яку можна розфарбувати граф. На письмі позначається як функція від графа: 𝛾(𝐺).

**4. Як визначаються нижня і верхня оцінки хроматичного числа?**



**5. Який граф називається планарним? У скільки кольорів його можна розфарбувати?**

Граф, який можна зобразити на площині так, що жодні два його ребра не перетинаються між собою, називають планарним. Будь-який планарний граф можна пофарбувати у 5 кольорів. Також у 1852 році була створена гіпотеза про чотири фарби, але вона не доведена.

**6. У скільки кольорів можна розфарбувати повний граф?**

Повний граф 𝐾𝑛 можна розфарбувати рівно 𝑛 фарбами

**7. Алгоритм простого перебору.**

Алгоритм простого перебору поступово перебирає усі вершини графа, та визначає їх колір (число) як найменший з невикористаних суміжними вершинами кольорів.

**8. Евристичний алгоритм розфарбовування графа.**

Евристичний алгоритм розфарбовування графа відрізняється від алгоритму просто перебору порядком перебору вершин. В еврестичному алгоритмі вершини перебираються в порядку зменшення ступеня вершин.

**9. Модифікований евристичний алгоритм розфарбовування графа.**

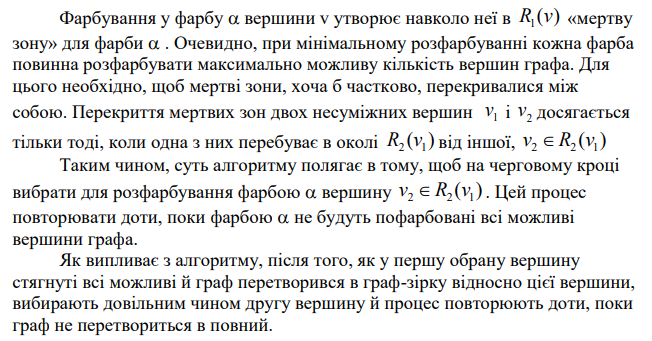
Проста модифікація описаної вище евристичної процедури базується на переупорядкуванні нерозфарбованих вершин по незростанню їх відносних степенів (степінь нерозфарбованих вершин у нерозфарбованому підграфі даного графа) . Дана модифікація полягає у тому, що якщо дві вершини мають однакові степені, то порядок таких вершин випадковий. Їх можна

впорядкувати по двокрокових степенях(сума відносних степенів суміжних

вершин у нерозфарбованому підграфі) . Двокроковий степінь визначимо як

суму відносних степенів суміжних вершин.

**10.Алгоритм розфарбування за методом А.П. Єршова.**



**11.Розфарбування за допомогою рекурсивної процедури послідовного розфарбування.**

1. Фіксуємо порядок обходу вершин.

2. Ідемо по вершинах, використовуючи такий найменший колір, який

не викликає конфліктів.

3. Якщо вже використаний колір вибрати неможливо, то тільки тоді

вводимо новий колір.

У процедурі використовується рекурсивний виклик процедури

фарбування наступної вершини у випадку успішного фарбування

попередньої вершини.

**12.«Жадібний» алгоритм розфарбування графів.**

1. Задаємо множину monochrom:= ∅ , куди будемо записувати всі

вершини, які можна пофарбувати одним кольором.

2. Переглядаємо всі вершини й виконуємо наступний «жадібний» алгоритм.Для початку розфарбування вибираємо вершину з номером 1 та розфарбовуємо її в колір 1 (червоний). Далі відбувається пошук несуміжної вершини з вершиною 1. Якщо така вершина знайдена, то вона також розфарбовується в колір 1 (червоний). Наступна знайдена для розфарбування кольором 1 вершина повинна бути не суміжною з двома попередніми. Процес продовжується до того часу, поки всі можливості розфарбувати вершини кольором 1 будуть вичерпані. Після цього вибираємо фарбу кольору 2 (синя) і розфарбовуємо нею вершину з мінімальним номером, яка є не розфарбованою до цього часу. Наступна вершина, яка підходить для розфарбування фарбою 2, повинна бути не суміжною з вершиною, яка була розфарбована кольором 2 (синій) на попередньому кроці. Процес розфарбування фарбою 2 також продовжується до того часу, поки не будуть вичерпані всі можливості розфарбування вершин цією фарбою. Перед вибором чергової фарби для розфарбування завжди перевіряємо, чи залишилися ще не розфарбовані вершини. Якщо такі вершини знайдено, то вибираємо чергову фарбу і продовжуємо процес розфарбування. Якщо ж всі вершини графа розфарбовано, то процес розфарбування жадібним алгоритмом закінчується