**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»**

Виконав:

студент групи IО-63

Братун Андрій

Залікова книжка № 6305

Перевірив Новотарський М. А.

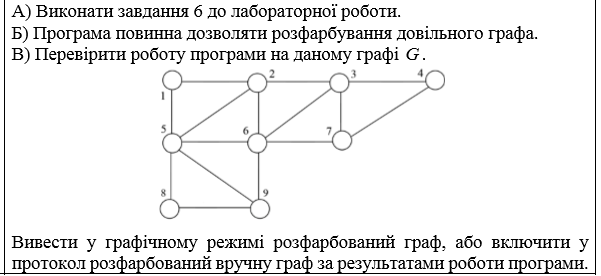
Київ 2016 р.

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

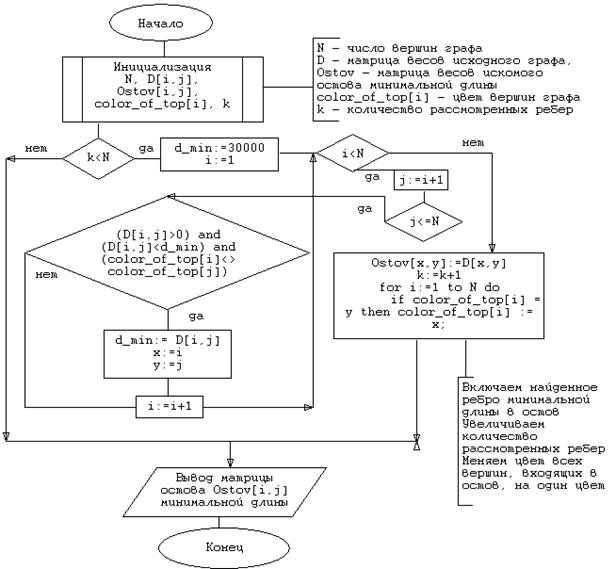
**Загальне завдання:** створити програму для правильного розфарбовування графа на основі одного з алгоритмів розфарбування.

**Завдання: Набути теоретичні знання по темі «Розфарбування графів». Створити програму розфарбування графів, яка реалізує «жадібний» алгоритм розфарбування.**

**Варіант виразу відповідно до індивідуального завдання – 6**

****

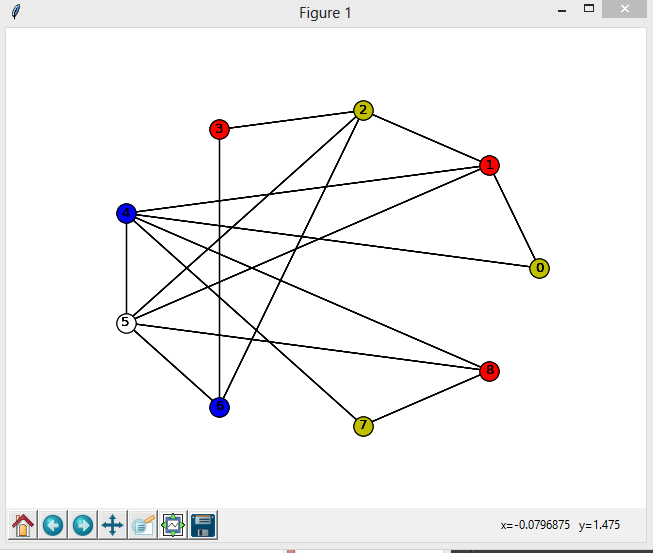
**Блок-схеми, які відповідають алгоритмам, що використані в лабораторній роботі.**

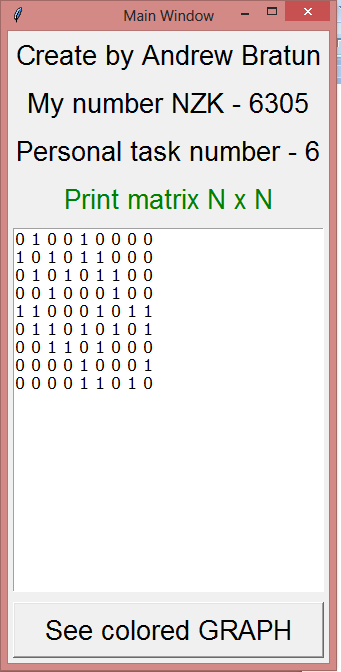
****

**Роздруківка того фрагменту тексту програми, який написаний індивідуально.**

**Module Main**

**import** networkx **as** nwx, matplotlib.pyplot **as** plt  
**from** tkinter **import** \*  
  
  
root = Tk()  
root.title("Main Window")  
  
#root.maxsize(width=320, height=190)  
#root.minsize(width=320, height=190)  
  
  
# My uniq task  
NZK = 6305  
number = NZK % 6 + 1  
task = "Personal task number - " + str(number)  
n = []  
**def get\_matrix**(n):  
 getter = tex1.get(1.0,END)  
 **for** i **in** getter.split('\n')[:-1]:  
 n.append([int(j) **for** j **in** i.split(" ")])  
 **return** n  
  
  
  
#раскраска графа  
**def graphColoring**():  
 **global** colored  
 colored=[[]]  
 color=0  
 stack=[]  
 num2=num1=-1  
 **def loop**(num):  
 **for** t **in** colored[color]:  
 **if** n[t][num]==1:  
 **return False  
 return True  
  
 for** a **in** n:  
 num1+=1  
 **if** num1 **not in** stack:  
 **if not** loop(num1):  
 colored.append([])  
 color+=1  
 stack.append(num1)  
 colored[color].append(num1)  
 num2=num1  
 **for** b **in** a[num1:]:  
 **if** b==0 **and** num2 **not in** stack **and** loop(num2):  
 stack.append(num2)  
 colored[color].append(num2)  
 num2+=1  
 **return** colored  
  
  
**def make\_graf2**(matrix,png,nodes):  
 fig = plt.figure()  
 graf = nwx.Graph()  
 **for** i **in** range(len(n)):  
 graf.add\_node(i)  
 **for** i **in** range(len(matrix)):  
 **for** j **in** range(len(matrix[1])):  
 **if** n[i][j] == 1:  
 graf.add\_edge(i, j)  
 nwx.draw(graf, pos = nwx.shell\_layout(graf))  
 list\_color = ['y','r','b','w','y']  
 **for** i **in** range(len(nodes)):  
 nwx.draw(graf, pos = nwx.shell\_layout(graf),node\_color = list\_color[i],nodelist = nodes[i],with\_labels=**True**)  
 plt.show()  
 fig.savefig(png)  
  
  
**def caalback**(event):  
 make\_graf2(get\_matrix(n), 'file.png', graphColoring())  
  
  
# #####################MAINLOOOOOOOP################  
  
# Info about student  
lab1 = Label(root, text='Create by Andrew Bratun', font='arial 20')  
lab2 = Label(root, text='My number NZK - 6305', font='arial 20')  
lab3 = Label(root, text=task, font='arial 20')  
lab4 = Label(root, text='Print matrix N x N', font='arial 20', foreground='green')  
tex1 = Text(root, width=10, height=20, font="Verdana 12", wrap=WORD)  
but1 = Button(root, text='See colored GRAPH', font='arial 20')  
but1.bind("<Button-1>", caalback)  
  
# Info about student  
  
lab1.grid(row=1, column=1, sticky=W+E+N+S, pady=5, padx=5)  
lab2.grid(row=2, column=1, sticky=W+E+N+S, pady=5, padx=5)  
lab3.grid(row=3, column=1, sticky=W+E+N+S, pady=5, padx=5)  
lab4.grid(row=4, column=1, sticky=W+E+N+S, pady=5, padx=5)  
  
tex1.grid(row=5, column=1, columnspan=2, sticky=W + E + N + S, pady=5, padx=5)  
  
but1.grid(row=6, column=1, sticky=W+E+N+S, pady=5, padx=5)  
  
  
  
  
root.mainloop()

**Роздруківка результатів виконання програми з контрольним прикладом **

****

**Аналіз результатів та висновки**

Протягом виконання роботи я вивчив основні методи роботи з графами, вивчив їх види та алгоритми пошуку шляхів, закріпив занання GUI в модулі tkinter в Python.

Складним завдання виявилося від лагодження програми на етапі розробки, оскільки виникпало багато логічних помилок які найскладніше виявити, проте завдання було цікавим, розробка своєї програми з інтерфейсом мені сподобалася. В ході розробки програми виникали деякі питання, але все вдалося знайти за допомогою додаткової інформації в інтернеті.