Министерство науки и высшего образования Российской Федерации

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

Кафедра информационных систем и программной инженерии

**Практическая работа № 3**

**«Топологическая сортировка»**

Выполнил:

ст. гр. ПРИ-120

К. А. Борисова

Принял:

Преподаватель

О. Н. Шамышева

Владимир, 2020

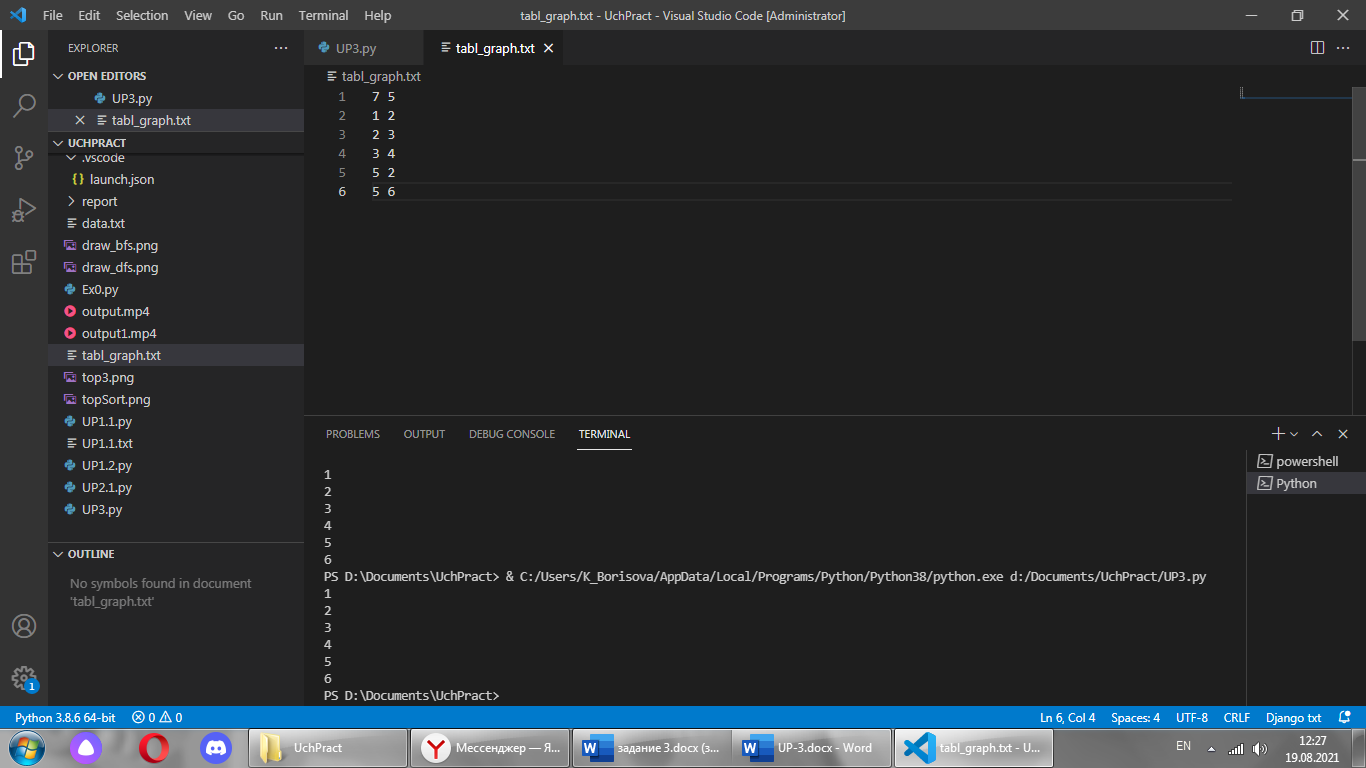
**Цель работы**

Научиться использовать алгоритм топологической сортировки в ориентированном графе.

**Ход работы**

Алгоритм получает на вход граф (информация считывается с файла). В результате топологической сортировки будет выведен такой линейный порядок на его вершинах, чтобы любое ребро вело от вершины с меньшим номером к вершине с большим номером. Очевидно, что если в графе есть циклы, то такого порядка не существует.

Листинг программы и информация из файла представлены ниже:

from PIL import Image, ImageDraw, ImageFont

from random import randint

videoDimensions = (1280, 720)

img = Image.new("RGB", videoDimensions, color = "white")

imgDrawer = ImageDraw.Draw(img)

font = ImageFont.truetype("arial.ttf", 16)

font\_text = ImageFont.truetype("arial.ttf", 30)

def graph\_from\_file():

   f = open("tabl\_graph.txt", "r")

   l = f.readline().rsplit()

   N = int(l[0])

   N\_R = int(l[1])

   A = [[0]\*N for i in range(N)]

   for i in range(1, N\_R + 1):

      l = f.readline().rsplit()

      A[int(l[0])][int(l[1])] = 1

   return A, N

def new\_coord\_1(X,Y):

   n = len(X)

   ni = int(n \*\* (1/2))

   nj = ni + 1

   if ni \*\* 2 == n:

      nj = ni

   elif ni\*nj < n:

      ni += 1

   a = (videoDimensions[0] - 100) // nj

   b = (videoDimensions[1] - 100) // ni

   x = 50

   y = 50

   z = 1

   for i in range(ni):

      for j in range(nj):

         print(z)

         X[z] = randint(x, x+a)

         Y[z] = randint(y, y+b)

         x = (x+a) % (videoDimensions[0] - 100)

         z += 1

         if z>=n:

            return X,Y

      y += b

def draw\_Graph(A,X,Y):

   for i in range(len(A)):

      for j in range(1, len(A)):

         if A[i][j] == 1:

            imgDrawer.line([(X[i], Y[i]), (X[j], Y[j])], "black", 2)

            imgDrawer.line([(X[j], Y[j]), (X[j] - 30, Y[j] + 20)], "black", 2)

            imgDrawer.line([(X[j], Y[j]), (X[j] - 30, Y[j] - 20)], "black", 2)

   for i in range(1, len(X)):

      imgDrawer.ellipse([(X[i] - 15, Y[i] - 15), (X[i] + 15, Y[i] + 15)], "white", "black", 2)

      imgDrawer.text((X[i] - 5, Y[i] - 5), str(i), (0,0,0), font)

def dfs(A,v, color, list):

   color[v] = 1

   for i in range(1, len(A)):

      if A[v][i] == 1 and color[i] == 0:

         dfs(A,i, color, list)

   k[0] += 1

   list[len(A)-k[0]] = v

   color[v] = 2

A, N = graph\_from\_file()

color = [0]\*N

list = [0]\*N

X = [0]\*N

Y = [0]\*N

new\_coord\_1(X,Y)

draw\_Graph(A,X,Y)

k = [0]

for v in range(1, N):

   if color[v] == 0:

      dfs(A, v, color, list)

imgDrawer.text((200,650), 'Результат: ', (0,0,0), font\_text)

for i in range(1, len(list)):

   imgDrawer.text((400 + i\*20,650), str(list[i]), (0,0,0), font\_text)

img.show()

img.save("top3.png", "PNG")

Результат работы программы:

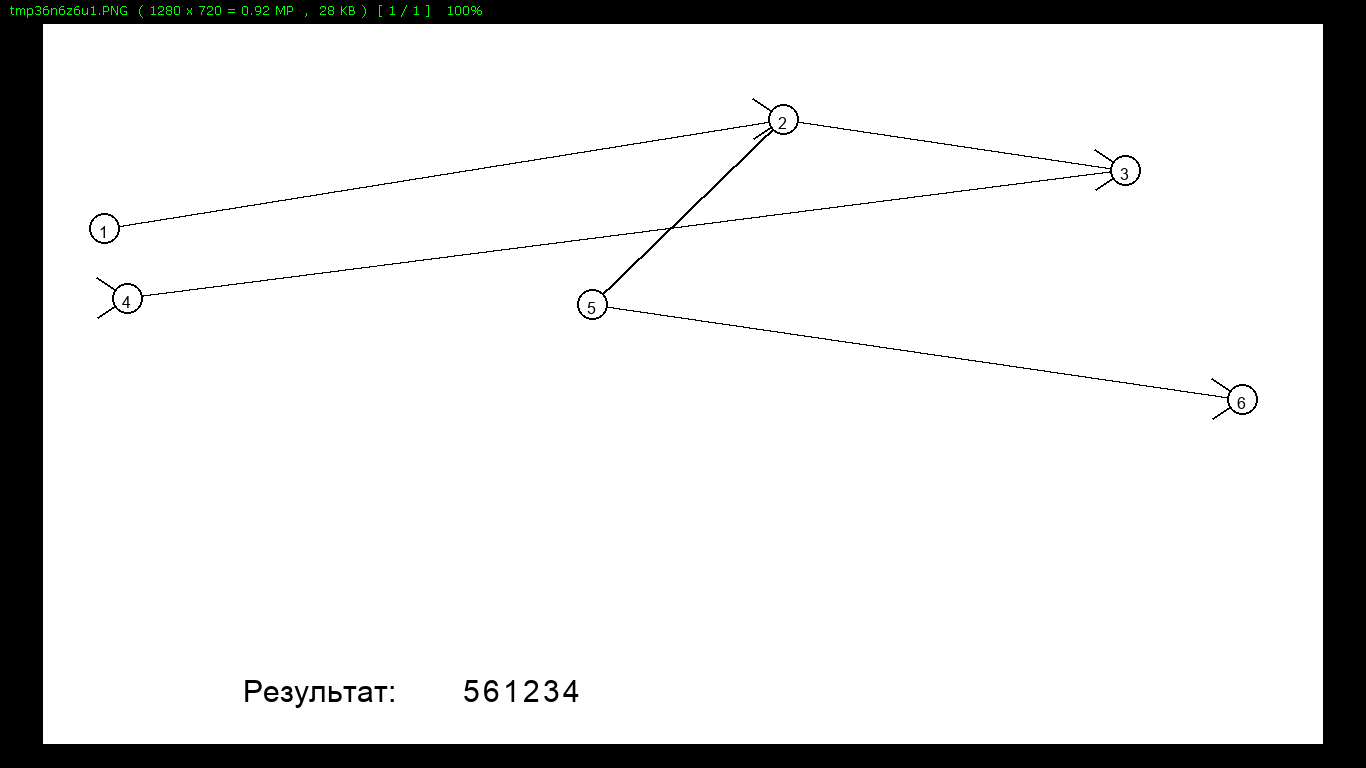
**Вывод**

Рисунок . Скриншот результата работы программы

В ходе данной работы была изучен алгоритм топологической сортировки и его применение в решении задач.