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 № 8**

**«Неориентированный невзвешенный граф.**

**Гамильтонов и Эйлеров циклы.»**

Выполнил:

ст. гр. ПРИ-120

К. А. Борисова

Принял:

Преподаватель

О. Н. Шамышева

Владимир, 2020

**Цель работы**

Изучить Гамильтонов и Эйлеров циклы. Реализовать Гамильтонов цикл.

**Ход работы**

**Эйлеровым циклом (путем)** графа называется цикл (путь), содержащий все ребра графа ровно один раз. Граф, обладающий эйлеровым циклом, называется **эйлеровым графом**.

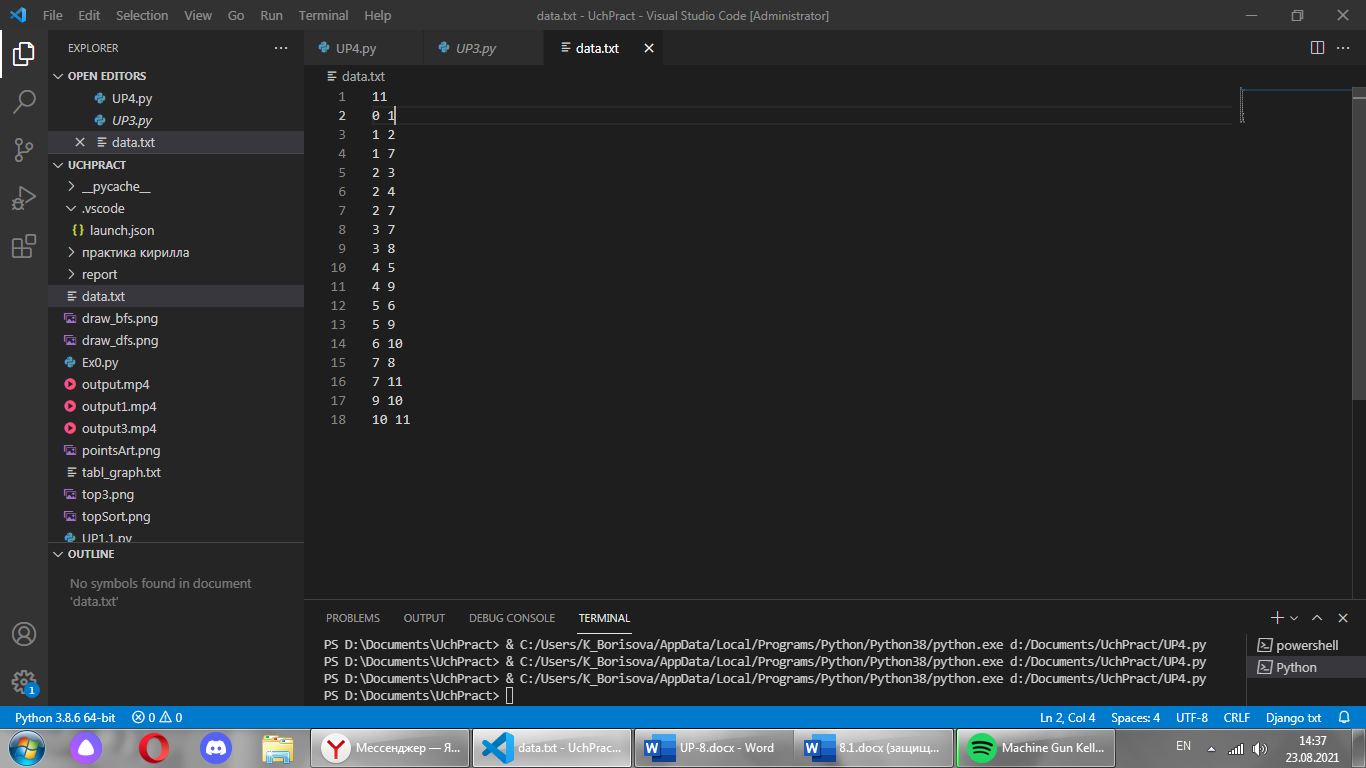
**Теорема 1.**  *Граф G обладает эйлеровым циклом с концами А и Б  тогда и только тогда, когда G  – связный и А, Б  –  единственные его вершины нечетной степени.*

**Теорема 2.** *Граф G является эйлеровым тогда и только тогда, когда G – связный и все его вершины имеют четную степень.*

**Гамильтоновым циклом (путем)** графа *G* называется цикл (путь), проходящий через каждую вершину *G* в точности по одному разу. Граф, обладающий гамильтоновым циклом, называется **гамильтоновым**. Критерий существования гамильтонова цикла в произвольном графе *G* еще не найден. Достаточным условием существования гамильтонова цикла является полнота графа *G*.

Алгоритм получает на вход граф (информация считывается с файла). В результате работы программы из исходного графа выделяются Гамильтонов цикл. Результат визуализирован на графе.

Листинг программы и информация из файла представлены ниже:

from PIL import Image, ImageDraw, ImageFont

from random import randint

videoDimensions = (1280, 1280)

img = Image.new(“RGB”, videoDimensions, color = ‘white’)

imgDrawer = ImageDraw.Draw(img)

font = ImageFont.truetype(‘C:\Windows\Fonts\Arial.ttf’, 30)

with open(‘data.txt’, ‘r’) as file:

    N = int(file.readline())

    A = [0] \* N

    for i in range(N):

        A[i] = [0] \* N

    while True:

        tmp = file.readline().rsplit()

        if tmp == []:

            break

        tmp[0], tmp[1] = int(tmp[0]) – 1, int(tmp[1]) – 1

        A[tmp[0]][tmp[1]] = 1

        A[tmp[1]][tmp[0]] = 1

def new\_coord():

    Y[0] = 10

    for i in range(N):

        X[i] = randint(80, videoDimensions[0] – 100)

        Y[i] = Y[i – 1] + videoDimensions[1] // N – 10

def draw\_graph(x, y):

    for i in range(N):

        for j in range(i + 1, N):

            if A[i][j] == 1:

                imgDrawer.line([(x[i], y[i]), (x[j], y[j])], ‘black’, 2)

    for i in range(1, N):

        imgDrawer.ellipse([(x[i] – 25, y[i] – 25), (x[i] + 25, y[i] + 25)], ‘white’, ‘black’, 2)

        imgDrawer.text((x[i] – 9, y[i] – 12), str(i), (0, 0, 0), font)

def draw\_hamilton():

    for i in range(len(path) – 1):

        imgDrawer.line([(X[path[i]], Y[path[i]]), (X[path[i + 1]], Y[path[i + 1]])], ‘red’, 4)

    for i in path:

        imgDrawer.ellipse([(X[i] – 25, Y[i] – 25), (X[i] + 25, Y[i] + 25)], ‘white’, ‘black’, 2)

        imgDrawer.text((X[i] – 9, Y[i] – 12), str(i), (0, 0, 0), font)

def GamCycle(gg):

    path.append(gg)

    if len(path) == N:

        if A[path[0]][path[-1]] == 1:

            return True

        else:

            path.pop()

            return False

    visited[gg] = True

    for next in range(N):

        if A[gg][next] == 1 and not visited[next]:

            if GamCycle(next):

                return True

    visited[gg] = False

    path.pop()

    return False

X = [0]\*N

Y = [0]\*N

visited = [False] \* N

path = []

new\_coord()

draw\_graph(X, Y)

GamCycle(0)

draw\_hamilton()

imgDrawer.text((100, 0), ‘Way: ‘, (0,0,0), font)

imgDrawer.text((100, 50), str(path), (0,0,0), font)

img.show()

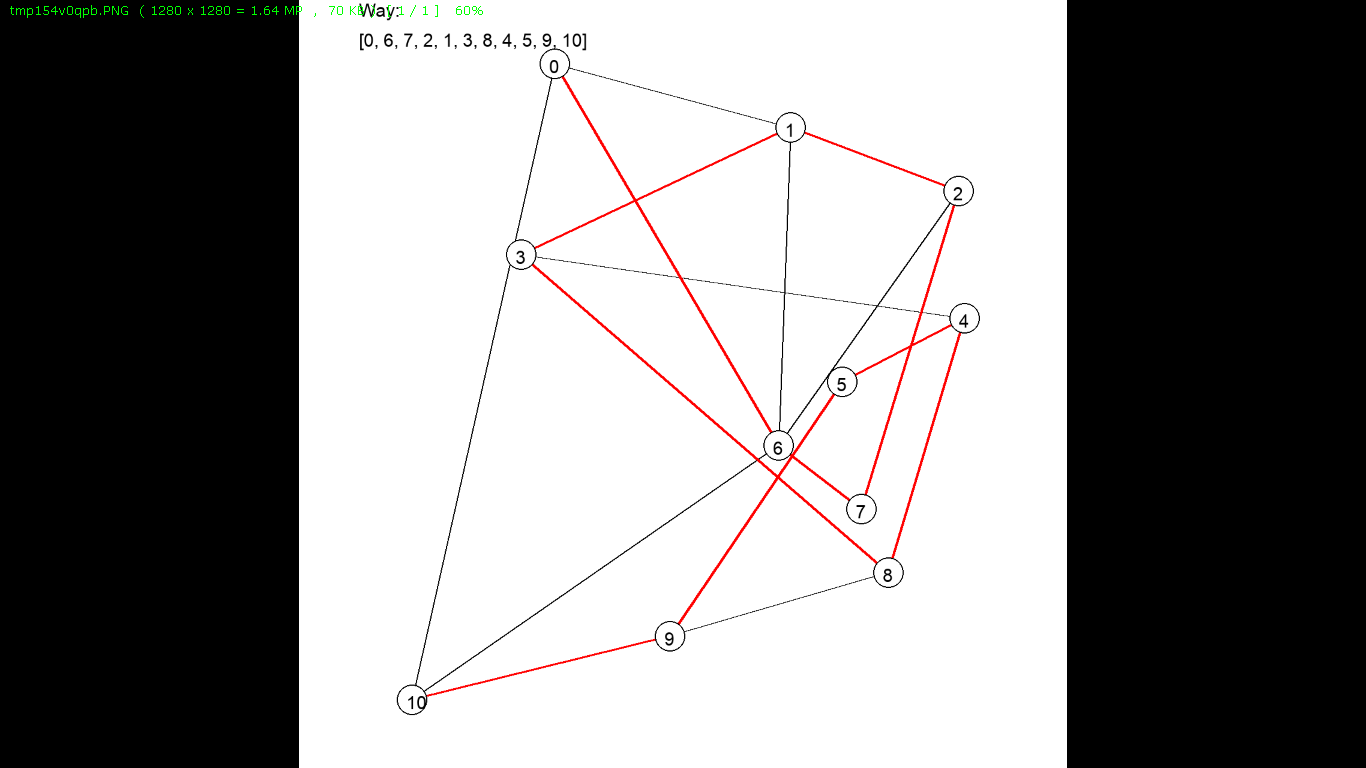


Рисунок . Гамильтонов цикл

**Вывод**

В ходе данной работы были изучены циклы: Гамильтонов и Эйлеров, – рассмотрен и реализован алгоритм поиск Гамильтонова цикла на практике.