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 № 5**

**«Волновой алгоритм»**

Выполнил:

ст. гр. ПРИ-120

К. А. Борисова

Принял:

Преподаватель

О. Н. Шамышева

Владимир, 2020

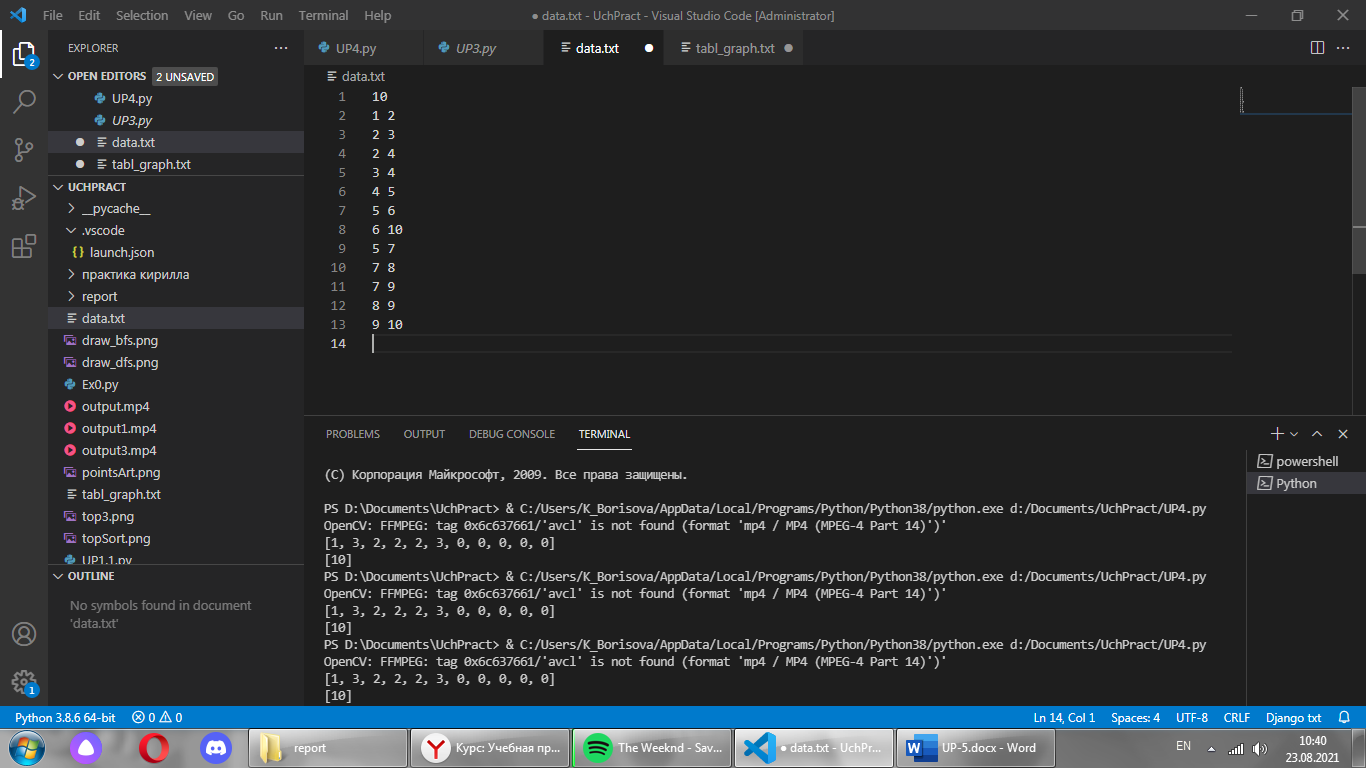
**Цель работы**

Научиться использовать волновой алгоритм поиска кратчайшего пути в графе.

**Ход работы**

Алгоритм получает на вход граф (информация считывается с файла). В результате работы программы будет выведен кратчайший путь от вершины «1» до вершины «10». Результат будет визуализирован в качестве графа с отмеченным синим цветом кратчайшим путем.

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)

with open('data.txt', 'r') as file:

    N = int(file.readline())

    A = []

    for i in range(N):

        A.append([])

    while True:

        tmp = file.readline().rsplit()

        if tmp == []:

            break

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

        A[tmp[0]].append(tmp[1])

        A[tmp[1]].append(tmp[0])

def new\_coord(X,Y):

    n = len(X)

    Y[0] = 10

    for i in range(n):

        X[i] = randint(50, videoDimensions[0] - 100)

        Y[i] = Y[i - 1] + videoDimensions[1] // n - 10

    return X, Y

def draw\_graph(x, y):

    for i in range(N):

        for j in A[i]:

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

    for i in range(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 + 1), (0, 0, 0), ImageFont.truetype('C:\Windows\Fonts\Arial.ttf', 30))

def draw\_route():

    for i in range(len(route) - 1):

        imgDrawer.line([(X[route[i]], Y[route[i]]), (X[route[i + 1]], Y[route[i + 1]])], 'blue', 4)

    for i in route:

        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 + 1), (0, 0, 0), ImageFont.truetype('C:\Windows\Fonts\Arial.ttf', 30))

def wave(d):

    used[d] = True

    for i in A[d]:

        if times[i] == 0:

            times[i] = times[d] + 1;

            if i == N - 1: return;

    for i in A[d]:

        if not(used[i]):

            wave(i)

def restore(d):

    k = times[d]

    for i in A[d]:

        if times[i] == k - 1:

            route.append(i)

            restore(i)

X = [0]\*N

Y = [0]\*N

used = [False]\*N

times = [0]\*N

times[0] = 1

route = []

route.append(N - 1)

wave(0)

restore(N - 1)

new\_coord(X, Y)

draw\_graph(X, Y)

draw\_route()

img.show()

Результат работы программы:

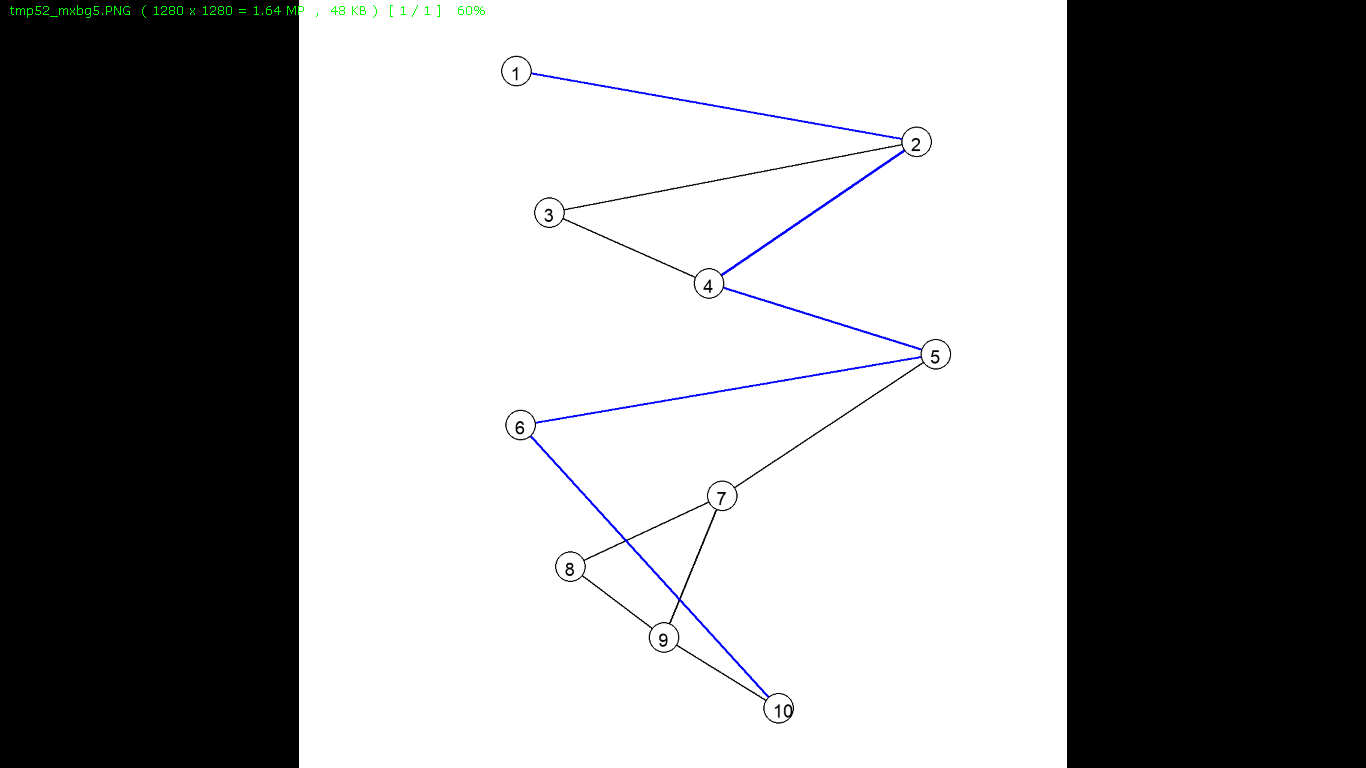


Рисунок . Скриншот работы программы

**Вывод**

В ходе работы был изучен и применен волновой алгоритм в задачах нахождения кратчайшего пути в графе.