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 № 1.1**

**«Чтение графа из файла. Рисование графа. DFS»**

**Выполнил**:

ст. гр. ПРИ-120

Д. А. Грачев

**Принял**:

Преподаватель

О. Н. Шамышева

Владимир, 2021

ЦЕЛЬ РАБОТЫ

Реализовать алгоритм обхода графа DFS, визуализировать граф на языке программирования Python.

ВЫПОЛНЕНИЕ РАБОТЫ

Листинг программы:

from PIL import Image, ImageDraw, ImageFont

from random import randint

import numpy as np

import cv2

videoDimensions = (1280, 720)

videoFps = 75

videoCodec = "avc1"

videoFilename = "output.mp4"

videoLength = 3

videoFourcc = cv2.VideoWriter\_fourcc(\*videoCodec)

video = cv2.VideoWriter(videoFilename, videoFourcc, videoFps, videoDimensions)

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

imgDrawer = ImageDraw.Draw(img)

font = ImageFont.truetype("impact.ttf", 18) #шрифт для цифр

font\_text = ImageFont.truetype("impact.ttf", 30) #заголовки

#Чтение графа из файла

*def* graph\_from\_file():

    file = open("data.txt", "r")

    kolvo = *int*(file.readline())

    graf = [[0]\*kolvo for i in range(kolvo)]

    for i in range(1, kolvo):

        l = file.readline().rsplit()

        graf[*int*(l[0])][*int*(l[1])] = 1

        graf[*int*(l[1])][*int*(l[0])] = 1

    return graf, kolvo

#Создание вершин графа

*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

#Рисование графа

*def* draw\_Graph(*A*,*X*,*Y*):

    for i in range(len(A)):

        for j in range(i+1, len(A)):

            if A[i][j] == 1:

                frame = img.copy()

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

                frameDrawer = ImageDraw.Draw(frame)

                video.write(cv2.cvtColor(np.array(frame), cv2.COLOR\_RGB2BGR))

    for i in range(1, len(X)):

        frame = img.copy()

        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*(5), (0,0,0), font)

        frameDrawer = ImageDraw.Draw(frame)

        video.write(cv2.cvtColor(np.array(frame), cv2.COLOR\_RGB2BGR))

#Обход графа dfs

*def* dfs(*A*,*v*,*color*,*p*,*z*):

    color[v] = 1

    z.append(v)

    for u in range(1, len(A)):

        if A[v][u] != 0 and color[u] == 0:

            p[u] = v

            dfs(A,u,color,p,z)

    color[v] = 2

#рисование линий, кругов и цифр

*def* draw\_dfs(*p*):

    for u in range(1, len(p)):

        if p[u] != 0:

            frame = img.copy()

            imgDrawer.line([(X[p[u]], Y[p[u]]), (X[u], Y[u])], "red", 5)

            frameDrawer = ImageDraw.Draw(frame)

            video.write(cv2.cvtColor(np.array(frame), cv2.COLOR\_RGB2BGR))

    for i in range(1, len(X)):

            frame = img.copy()

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

            imgDrawer.text((X[i] - 4, Y[i] - 10), *str*(i), (0,0,0), font)

            frameDrawer = ImageDraw.Draw(frame)

            video.write(cv2.cvtColor(np.array(frame), cv2.COLOR\_RGB2BGR))

A, N = graph\_from\_file()

color = [0]\*N

p = [0]\*N

d = [0]\*N

fi = [0]\*N

X = [0]\*N

Y = [0]\*N

z = []

new\_coord\_1(X,Y)

draw\_Graph(A,X,Y)

for v in range(1, len(A)):

    if color[v] == 0:

        dfs(A,v,color,p,z)

draw\_dfs(p)

frame = img.copy()

imgDrawer.text((100, 0), 'Graph: ', (0,0,0), font\_text)

frameDrawer = ImageDraw.Draw(frame)

video.write(cv2.cvtColor(np.array(frame), cv2.COLOR\_RGB2BGR))

for i in range(len(z)):

    frame = img.copy()

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

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

    frameDrawer = ImageDraw.Draw(frame)

    video.write(cv2.cvtColor(np.array(frame), cv2.COLOR\_RGB2BGR))

img.save("draw\_dfs.png", "PNG")

img.show()

video.release()

Скриншот работы программы и данные из файла data.txt представлены на Рис.1.1 и 1.2.

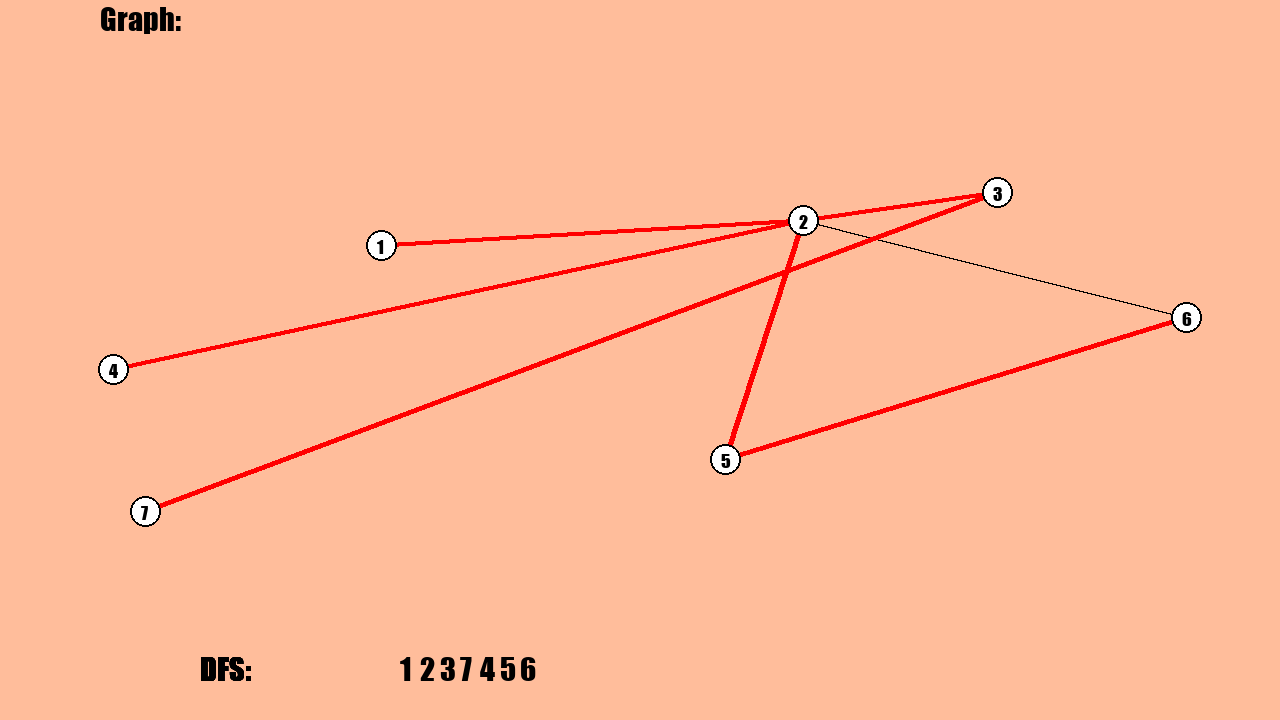
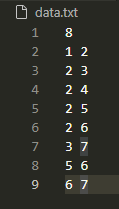
 

Рисунок 1.1 и 1.2. Результат работы программы и data.txt

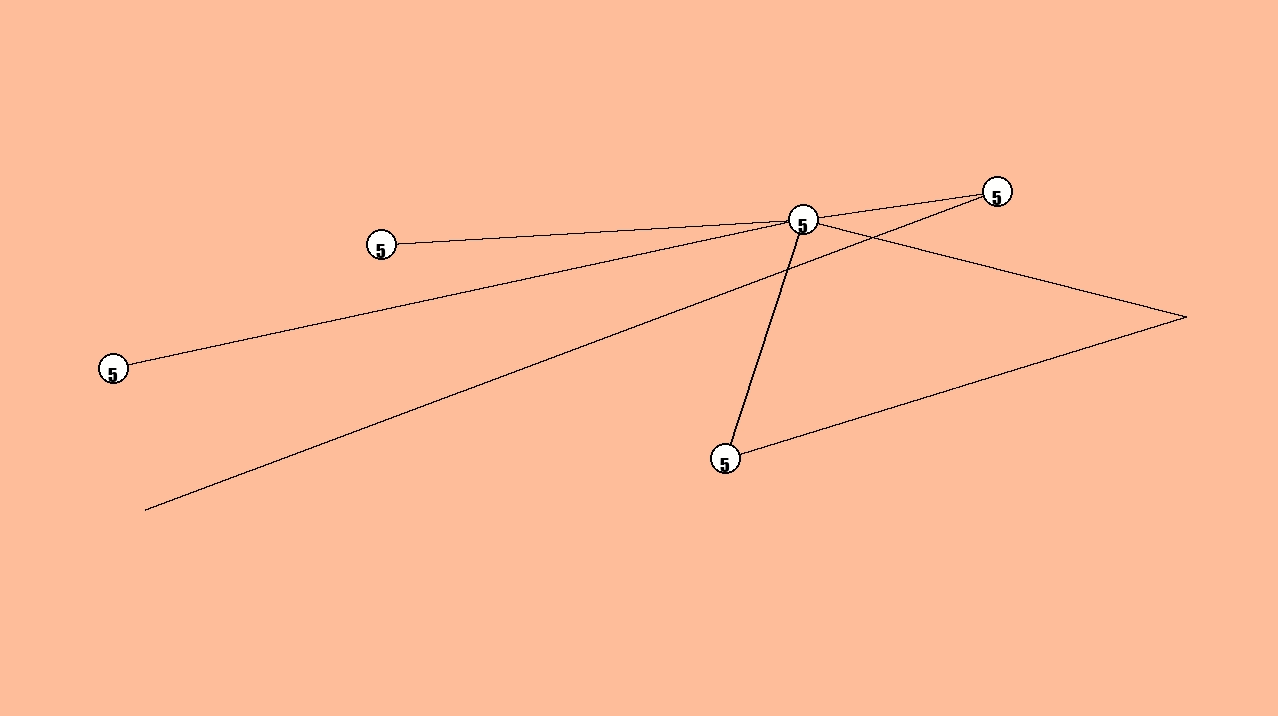
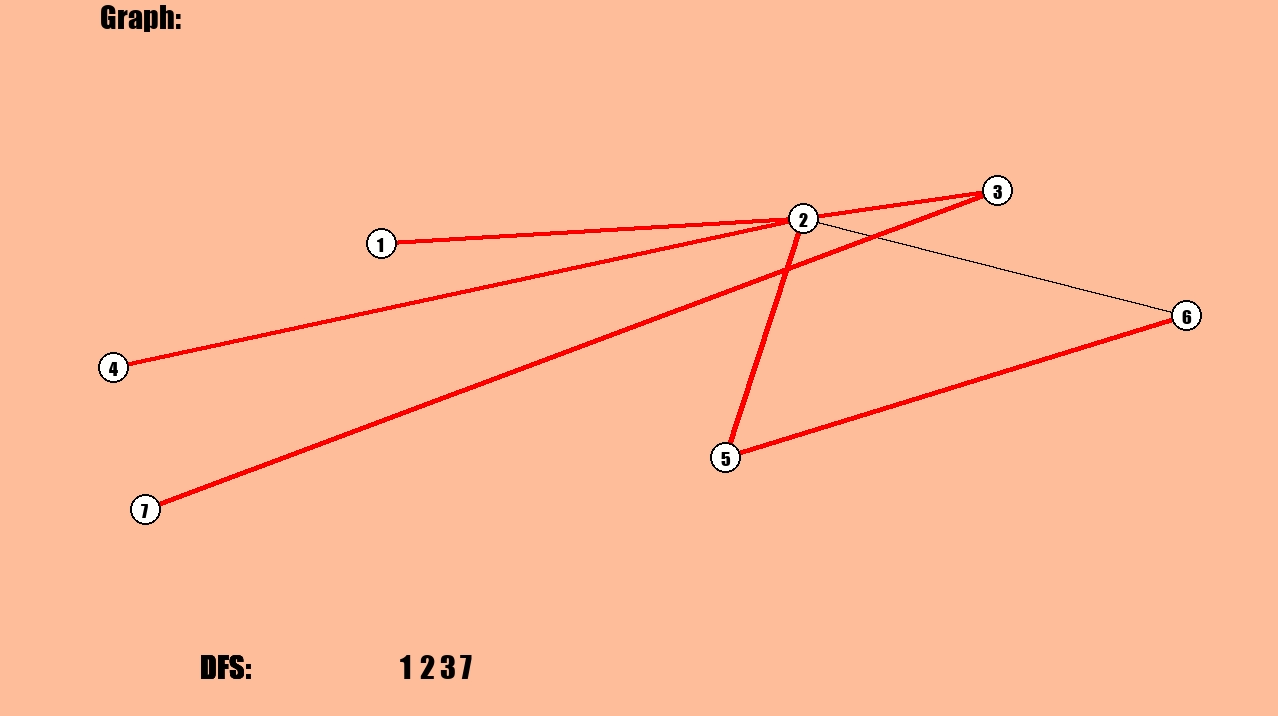
 

Рисунок 2. Скриншоты из видео

ВЫВОД

В ходе выполнения работы были получены практические навыки по реализации алгоритма обхода графа DFS, визуализирован граф на языке программирования Python.