 Министерство науки и высшего образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЁТ**

по лабораторной работе №7

по курсу «Логика и основы алгоритмизации в ИЗ»

на тему «Обход графа в глубину»

Выполнили:

студенты группы 22ВВВ2

Попов Д. А.

Приняли:

Д.т.н., профессорМитрохин М.А.

К.э.н. доцент Акифьев И.В.

Пенза 2023

**Название**

Обход графа в глубину

**Цель работы**

Научиться реализовывать обход графа в глубину

**Лабораторное задание**

**Задание 1**

1. Сгенерируйте (используя генератор случайных чисел) матрицу смежности для неориентированного графа *G*. Выведите матрицу на экран.
2. Для сгенерированного графа осуществите процедуру обхода в глубину, реализованную в соответствии с приведенным выше описанием.

**3.**\* Реализуйте процедуру обхода в глубину для графа, представленного списками смежности.

**Задание 2\***

1. Для матричной формы представления графов выполните преобразование рекурсивной реализации обхода графа к не рекурсивной.

**Листинг**

import random

graph = []

size = 5

sp\_graph = [[" "] \* size for i in range(size)]

#1 задание

defdfs(graph, start, visited=None):

if visited is None:

visited = set()

visited.add(start)

print(start)

for n in range(size):

if (graph[start][n] == 1) and (not (n in visited)):

dfs(graph, n, visited)

return visited

#2 задание

def dfs2(graph, start, visited=None):

if visited is None:

visited = set()

visited.add(start)

print(start)

for i in range(size):

if type(sp\_graph[start][i]) == int and (not (sp\_graph[start][i] in visited)):

dfs2(graph, int(sp\_graph[start][i]), visited)

return visited

#созданиематрицысмежностей

for i in range(size):

graph.append([])

for j in range(size):

graph[i].append(0)

for i in range(size):

for j in range(size):

f = random.randint(0,1)

graph[i][j] = f

graph[j][i] = f

for i in range(size):

for j in range(size):

print(f"{graph[i][j]}", end="")

print()

print()

#созданиеспискасмежностей

ch = 0

for i in range(size):

ch = 0

for j in range(size):

if (graph[i][j] == 1):

sp\_graph[i][ch] = j

ch += 1

for i in range(size):

for j in range(size):

print(f"{sp\_graph[i][j]}", end="")

print()

print()

dfs(graph, 0)

print()

dfs2(sp\_graph, 0)

**Вывод:**

В ходе выполнения лабораторной работы я научился реализовывать обход графа в глубину.

Ссылка на репозиторий: https://github.com/roc25/laba7\_logica