Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №4

по курсу «Логика и основы алгоритмизации в инженерных задачах»

на тему «Обход графа в глубину»

Выполнил:

студент группы 20ВВ2

Беляков А.М.

Проверили:

Митрохин М.А.

Юрова О.В.

Пенза 2021

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.

Задание 1:

Изображение выглядит как текст

Автоматически созданное описание

Листинг:

Graph.py

def dfs(self, index):  
 to\_ret = [self.\_matrix.dfs(self.\_nodes[index]), self.\_adjacency\_list.dfs(self.\_nodes[index])]  
 return to\_ret

Matrix.py

def dfs(self, node):  
 return self.\_dfs\_real(node)  
  
def \_dfs\_real(self, node):  
 visits = []  
 stack = [node.index]  
 while len(stack) > 0:  
 item = stack.pop()  
 if item not in visits:  
 visits.append(item)  
 for i in range(len(self.\_matrix[item]) - 1, 0, -1):  
 if len(self.\_matrix[item][i]) > 0:  
 stack.append(i)  
 return visits

AdjacencyList.py

def dfs(self, node):  
 visits = []  
 return self.\_dfs\_real(node, visits)  
  
def \_dfs\_real(self, node, visits):  
 visits.append(node.index)  
 for i in self.\_adj\_list[node]:  
 if i[0].index not in visits:  
 self.\_dfs\_real(i[0], visits)  
 return visits

Вывод: во время реализации данной лабораторной работы мы научились создавать рекурсивный и не рекурсивный алгоритм поиска в глубину по графу.