



Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУК «Информатика и управление»

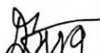
КАФЕДРА ИУК4 «Программное обеспечение ЭВМ, информационные технологии»

ЛАБОРАТОРНАЯ РАБОТА №5

«Остовы графов»

ДИСЦИПЛИНА: «Дискретная математика»

Выполнил: студент гр. ИУК4-32Б


(Подпись)

(____ Зудин Д.В. ____)
(Ф.И.О.)

Проверил:


(Подпись)

(____ Никитенко У.В. ____)
(Ф.И.О.)

Дата сдачи (защиты): 19.12.2022

Результаты сдачи (защиты):

- Балльная оценка: 75

- Оценка: хорошо

Калуга, 2022 г.

Цель: изучение алгоритмов обхода графов и алгоритмов поиска остова.

Задачи:

1. Построение остова;
2. Поиск в ширину и поиск в глубину.

Вариант №4

Формулировка общего задания

Для графа G_4 , полученного в лабораторной работе №4, построить остов. Четные варианты – методом обхода в ширину, нечетные варианты – методом обхода в глубину.

Результат выполнения программы для общего задания

```
[0, 1, 0, 0, 0, 0, 0, 0, 0, 0]
[1, 0, 1, 0, 0, 0, 0, 0, 0, 0]
[0, 1, 0, 1, 0, 0, 0, 0, 0, 0]
[0, 0, 1, 0, 1, 0, 0, 0, 0, 0]
[0, 0, 0, 1, 0, 1, 0, 0, 0, 0]
[0, 0, 0, 0, 1, 0, 1, 0, 0, 0]
[0, 0, 0, 0, 0, 1, 0, 1, 0, 0]
[0, 0, 0, 0, 0, 0, 1, 0, 1, 0]
[0, 0, 0, 0, 0, 0, 0, 1, 0, 1]
[0, 0, 0, 0, 0, 0, 0, 0, 1, 0]
```

Листинг программы для общего задания

```
class Graph:

    def __init__(self, adj):
        self.adj = adj
        self.v = len(adj)
        self.vlist = []

    def add_edge(self, a, b):
        self.adj[a][b] = 1
        self.adj[b][a] = 1
        if a == b:
            self.adj[a][b] = 2

    def BFS(self, start):
        visited = [False] * self.v
```

```

q = [start]
visited[start] = True

while q:
    vis = q[0]
    self.vlist.append(vis)
    q.pop(0)
    for i in range(self.v):
        if (self.adj[vis][i] != 0 and
            (not visited[i])):
            q.append(i)
            visited[i] = True

G_1 = [[0, 0, 0, 1, 0, 0, 0, 0, 0, 0],
        [0, 0, 0, 0, 1, 0, 0, 0, 0, 0],
        [0, 0, 0, 0, 0, 0, 1, 1, 0, 0],
        [1, 0, 0, 0, 0, 0, 0, 0, 0, 1],
        [0, 1, 0, 0, 0, 1, 0, 0, 0, 0],
        [0, 0, 0, 0, 1, 0, 0, 1, 1, 0],
        [0, 0, 1, 0, 0, 0, 0, 1, 1, 0],
        [0, 0, 1, 0, 0, 1, 1, 0, 0, 0],
        [0, 0, 0, 0, 0, 1, 1, 0, 0, 0],
        [0, 0, 0, 1, 0, 0, 0, 0, 0, 0], ]

dop_G_1 = G_1

for i in range(len(G_1)):
    for j in range(len(G_1)):
        e = 1
        if i == j:
            e = 2
        if G_1[i][j] == e:
            dop_G_1[i][j] = 0
        if G_1[i][j] == 0:
            dop_G_1[i][j] = e

G_0 = [[0 for _ in range(len(G_1))] for _ in range(len(G_1))]

a = Graph(dop_G_1)
b = Graph(G_0)

```

```
a.BFS(0)

for i in range(len(a.vlist) - 1):
    b.add_edge(a.vlist[i], a.vlist[i + 1])

print(*b.adj, sep='\n')
```

Выводы:

В ходе работы были изучены основы теории графов, базовые понятия и определения; освоены компьютерные способы представления графов и операции над ними.