



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 **М-КФ «Машиностроительный»**

КАФЕДРА **М10-КФ «Высшая математика и физика»**

ЛАБОРАТОРНАЯ РАБОТА №5

«Обходы графа в глубину и ширину»

ДИСЦИПЛИНА: «Дискретная математика»

Выполнил: студент гр. ИУК4-32Б _____ (Карельский М.К.)
(Подпись)

Проверил: доцент кафедры М10-КФ _____ (Булычев В.А.)
(Подпись)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:
- Оценка:

Калуга, 2021

Цель: реализация алгоритмов обхода простого неориентированного графа в глубину и ширину и поиск с их помощью путей между двумя заданными вершинами.

Задание:

1. Прочитайте из файла «data.txt» значения переменных N, source, target и матрицу смежности A размера $N \times N$ для своего варианта.
2. Составьте программу для обхода графа в глубину и найдите с её помощью путь из вершины source в вершину target и его длину или сообщите, что такого пути нет.
3. Составьте программу для обхода графа в ширину и найдите с её помощью кратчайший путь из вершины source в вершину target и его длину или сообщите, что такого пути нет.
4. Нарисуйте на графе пути, найденные в пунктах 2 и 3. Этот пункт задания можно сделать как с помощью компьютера, так и вручную.

Вариант 10

Листинг:

```
n = int
source = int
target = int
A = []

with open('data.txt') as f:
    n, source, target = f.readline().split()
    n = int(n)
    source = int(source)
    target = int(target)

    print('  ', ''.join('{:>3}'.format(i + 1) for i in range(n)))
    print()
    for line in f:
        numbers = line.split()
        A.append([int(numbers[i]) for i in range(n)])
        print('{:>2}'.format(len(A)), ''.join('{:>3}'.format(i)
        for i in numbers))
    print()

Mark = [False]*n
isFound = False
route = []

def DFS(u):
    global isFound
    route.append(u + 1)
    Mark[u] = True
    if u + 1 == target:
        isFound = True
    if not isFound:
```

```

        for v in range(n):
            if A[u][v] == 1 and not Mark[v]:
                DFS(v)
                if isFound:
                    return
                else:
                    route.pop()

DFS(source-1)
print('Путь из {} в {} в глубину: '.format(source, target), end =
'')
if isFound:
    print(' -> '.join(str(i) for i in route))
    print('Длина =', len(route) - 1)
else:
    print('не найден')

Mark = [False]*n
Q = [source - 1]
Mark[source - 1] = True
head = 0
elementsInTier = [1, 0]
tiers = [[source], []]
currentTier = 0

while head < len(Q):
    u = Q[head]
    elementsInTier[currentTier] -= 1
    for v in range(n):
        if A[u][v] == 1 and not Mark[v]:
            Mark[v] = True
            elementsInTier[currentTier + 1] += 1
            tiers[currentTier + 1].append(v + 1)
            Q.append(v)
    head += 1
    if elementsInTier[currentTier] == 0:
        currentTier += 1
        elementsInTier.append(0)
        tiers.append([])
    tiers.pop()
    tiers.pop()

i = 0
while i < len(tiers) and not target in tiers[i]:
    i += 1

print('Путь из {} в {} в ширину: '.format(source, target), end =
'')
if i < len(tiers):
    route = [0] * (i + 1)
    route[i] = target
    while i > 0:
        for t in tiers[i - 1]:
            if A[t - 1][route[i] - 1]:
                route[i - 1] = t

```

```

        break
    i -= 1

    print(' -> '.join(str(i) for i in route))
    print('Длина =', len(route) - 1)
else:
    print('не найден')

```

Результат:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
4	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
5	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
6	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0
7	0	1	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1
8	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1
9	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
10	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1
12	0	0	0	1	0	0	0	1	1	0	0	0	0	0	1	0	0
13	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
15	0	0	1	0	0	1	0	0	0	0	0	1	0	1	0	0	0
16	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0
17	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0

Путь из 2 в 10 в глубину: 2 -> 5 -> 7 -> 11 -> 16 -> 8 -> 10
 Длина = 6
 Путь из 2 в 10 в ширину: 2 -> 7 -> 17 -> 8 -> 10
 Длина = 4

Рис. 1. Результат работы

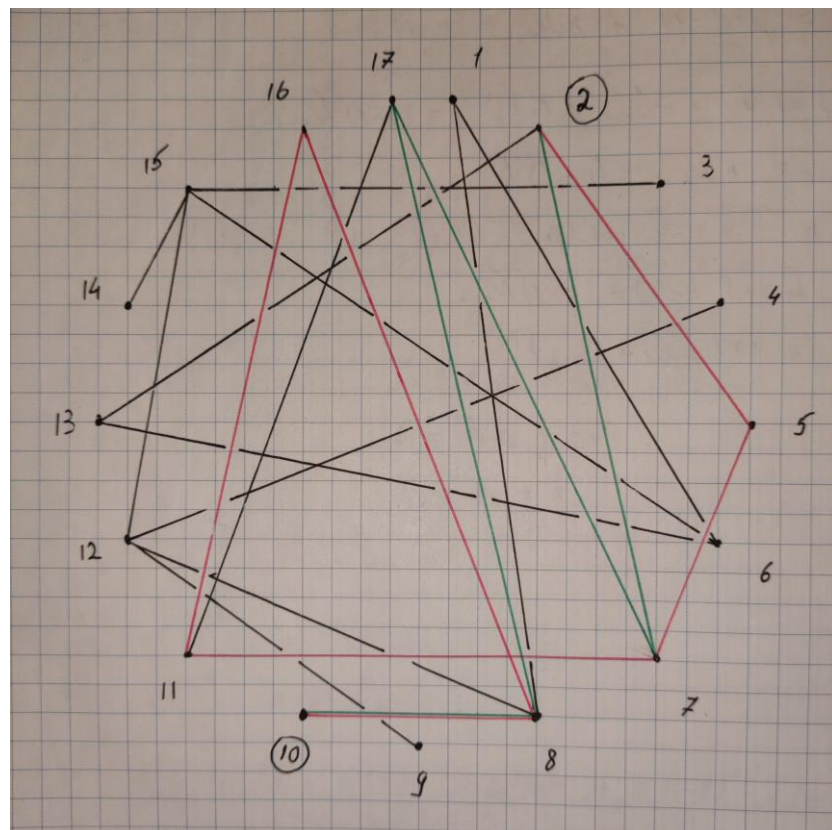
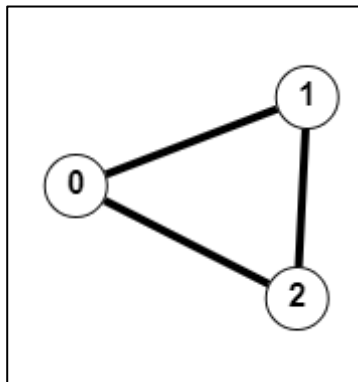


Рис. 2. Граф путей

Вывод: в ходе выполнения лабораторной работы составлены программы, осуществляющие обход простого неориентированного графа в глубину и ширину, и поиск с их помощью пути между двумя заданными вершинами.

Контрольные вопросы:

1. Какой граф называется простым?
Граф без ребер и петель
2. Какой граф называется неориентированным? Каким свойством обладает его матрица смежности?
Если все ребра графа имеют оба направления, то граф называется неориентированным. Матрица смежности такого графа будет симметричной
3. Что такое путь на графе? Что называется длиной пути?
Путь – последовательность смежных вершин. Длина пути – количество ребер, соединяющих вершины последовательности
4. Приведите пример графа и двух его вершин, для которых обход в глубину находит



- a) Кратчайший путь
Из 0 в 1
- b) Не кратчайший путь
Из 0 в 2