|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **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 №6**

**«Кратчайшие пути на взвешенном графе»**

**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 различных алгоритмов для поиска расстояний и кратчайших путей на взвешенном графе.

**Задание:**

1. Прочитайте из файла «data.txt» значения переменных N, source, target и весовую матрицу A размера для своего варианта.
2. С помощью алгоритма Форда-Беллмана найдите расстояния от вершины source до всех остальных вершин графа.
3. С помощью алгоритма Дейкстры найдите расстояние и кратчайший путь от вершины source до вершины target. Нарисуйте граф и найденный путь.
4. С помощью алгоритма Флойда-Уоршалла найдите матрицу попарных расстояний между всеми вершинами графа.
5. Нарисуйте на графе путь, найденный в пункте 3. Этот пункт задания можно сделать как с помощью компьютера, так и вручную.

**Вариант 10**

**Листинг:**

import math

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()

for i in range(n):

for j in range(n):

if i == j:

A[i][j] = 0

elif A[i][j] == 0:

A[i][j] = math.inf

d = A[source-1].copy()

for k in range(n-2):

for v in range(n):

for u in range(n):

d[v] = min(d[v],d[u]+A[u][v])

print('Алгоритм Форда-Беллмана:')

print(d)

Mark = [False]\*n

d = [math.inf]\*n

d[source-1] = 0

prev = [None]\*n

for i in range(n-1):

m = math.inf

for v in range(n):

if (not Mark[v]) and (d[v]<m):

m = d[v]

u = v

Mark[u] = True

for v in range(n):

if (not Mark[v]) and (d[u]+A[u][v]<d[v]):

d[v] = d[u] + A[u][v]

prev[v] = u

Path = [target-1]

v = target-1

while v!=source-1:

v = prev[v]

Path.insert(0,v)

print('Алгоритм Дейкстры:')

print('Путь: ', ' -> '.join(str(i + 1) for i in Path))

print('Расстояние =', d[target-1])

for k in range(n):

for u in range(n):

for v in range(n):

A[u][v] = min(A[u][v], A[u][k]+A[k][v])

print('Алгоритм Флойда-Уоршалла:')

print(' ', ''.join('{:>4}'.format(i + 1) for i in range(n)))

print()

for i in range(n):

print('{:>2}'.format(i + 1), ''.join('{:>4}'.format(a) for a in A[i]))

**Результат:**

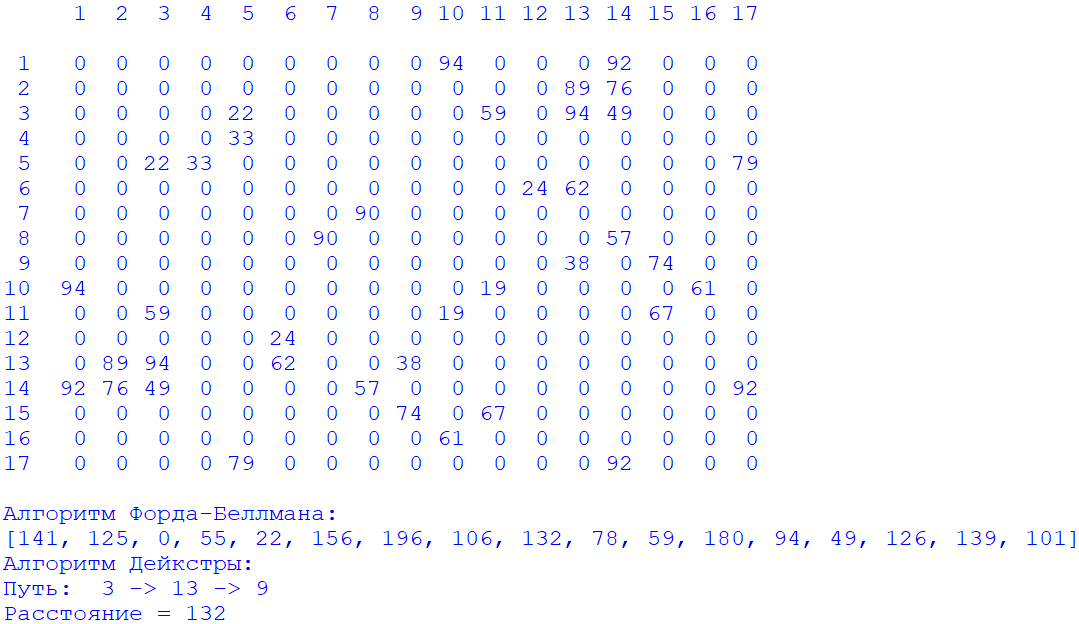


Рис. 1.1. Результат работы

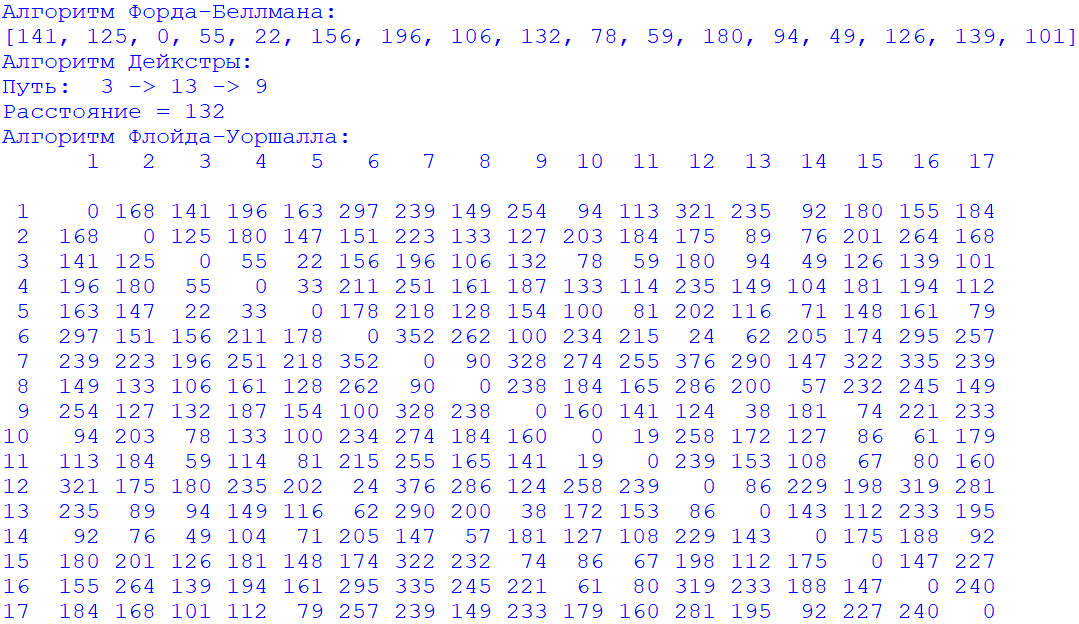


Рис. 1.2. Результат работы

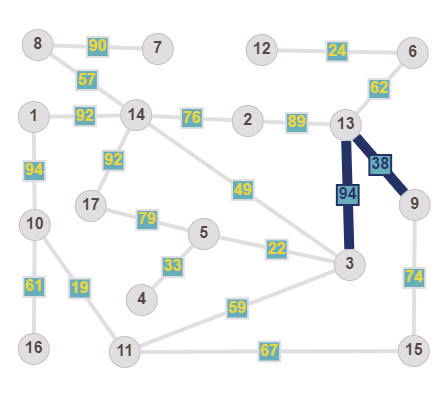


Рис. 2. Алгоритм Дейкстры

**Вывод:** в ходе выполнения лабораторной работы составлены программы, реализующие алгоритмы Форда-Беллмана, Дейкстры и Флойда-Уоршалла для поиска кратчайших путей на взвешенном графе.

**Контрольные вопросы:**

1. Нарисуйте пример графа и приведите на нём примеры следующих понятий: путь, цепь, простая цепь, цикл, простой цикл.

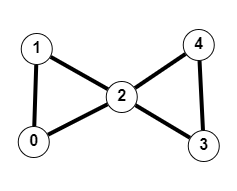


Рис. 3. Пример графа

Путь – последовательность вершин 0, 1, 2, 1

Цепь – последовательность вершин 0, 2, 4, 3, 2

Простая цепь – последовательность вершин 0, 1, 2, 3, 4

Цикл – последовательность вершин 0, 1, 2, 3, 4, 2, 0

Простой цикл – последовательность вершин 0, 1, 2, 0

1. Что называется длиной пути на не взвешенном графе? На взвешенном?

На не взвешенном – количество ребер пути

На взвешенном – сумма весов ребер пути

1. Какова сложность алгоритма Форда-Беллмана? Какова сложность алгоритма Дейкстры? Какова сложность алгоритма Флойда-Уоршалла?

Алгоритм Форда-Беллмана – O(p3)

Алгоритм Дейкстры – O(p2)

Алгоритм Флойда-Уоршалла – O(p3)