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

|  |  |
| --- | --- |
| **ФАКУЛЬТЕТ** | "Фундаментальные науки" |
| **КАФЕДРА** | "Высшая математика" |



**О Т Ч Е Т**

ЛАБОРАТОРНАЯ РАБОТА №6

|  |  |  |
| --- | --- | --- |
| **ДИСЦИПЛИНА:** | | "Дискретная математика" |
| **ТЕМА:** | "Кратчайшие пути на графе" | |

|  |  |
| --- | --- |
| Выполнил: студент гр. ИТД.Б-31 | Лысенков А.Е. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Проверил: доцент кафедры ФН3-КФ | Булычев В.А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

|  |  |  |
| --- | --- | --- |
| Дата сдачи (защиты) лабораторной работы: | |  |
| Результаты сдачи (защиты):  Количество рейтинговых баллов |  | |
|  |  | |

Калуга, 2015 г

**Цель работы:**

1. Знакомство с возможностями пакета GraphTheory.
2. Реализация алгоритмов Дейкстры и Флойда-Уоршелла для поиска кратчайшего пути на взвешенном графе.

**Задание:**

1. Прочитайте из файла «data6.txt» весовую матрицу A исходного графа G для вашего варианта.
2. В графе G, заданном весовой матрицей A, найдите кратчайшие пути и их длины от вершины 1 до всех остальных вершин графа, используя алгоритм Дейкстры.
3. С помощью алгоритма Флойда-Уоршелла найдите матрицу попарных расстояний между всеми вершинами графа. Сравните с результатами, полученными в п.2.
4. Подключите пакет **GraphTheory**. Выведите на экран изображение графа с помощью DrawGraph(). Проверьте результаты пунктов 2 и 3 с помощью DijkstrasAlgorithm() и AllPairsDistance().

В отчёт включите весовую матрицу; две программы с выводом результатов; результаты, полученные с помощью пакета GraphTheory.

**Теоретическая часть:**

1. ***Дайте определение следующих понятий: путь, цепь, простая цепь, цикл, простой цикл.***

*Путь – любая последовательность смежных вершин графа. Цепь – путь, все ребра которого различны. Простая цепь – цепь, все вершины которой кроме, возможно, первой и последней различны. Цикл - замкнутая цепь. Простой цикл – замкнутая простая цепь.*

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

*Длиной пути на невзвешенном графе называется число ребер его составляющих.*

*Длиной пути на взвешенном графе называется сумма весов составляющих его ребер.*

1. ***Какова сложность алгоритма Дейкстры? Какова сложность алгоритма Флойда-Уоршелла?****;*

**Код программы:**

read "data.txt":

with(GraphTheory):

n := sqrt(ArrayNumElems(A)):

tmp := copy(A):

for i from 1 to n do

for j from 1 to n do

if (tmp[i, j] = infinity) then

tmp[i, j] := 0:

end if:

end do:

end do:

G := Graph(tmp):

DrawGraph(G):

FloydMarshall := proc(n)

global PinkFloydMatrix:

local k, i, j:

for k from 1 to n do

for i from 1 to n do

for j from 1 to n do

if (PinkFloydMatrix[i, k] + PinkFloydMatrix[k, j] < PinkFloydMatrix[i, j]) then

PinkFloydMatrix[i, j] := PinkFloydMatrix[i, k] + PinkFloydMatrix[k, j]:

end if:

end do:

end do:

end do:

end proc:

Dijkstra := proc(n, s, t)

global DijkstraMatrix:

local minPath, isPassed, path, i, j, index, min:

minPath := [seq(infinity, i = 1..n)]:

isPassed := [seq(0, i = 1..n)]:

path := [seq(0, i = 1..n)]:

minPath[s] := 0:

for i from 1 to n - 1 do

min := infinity:

for j from 1 to n do

if ((isPassed[j] = 0) and (minPath[j] < min)) then

min := minPath[j]:

index := j:

end if:

end do:

for j from 1 to n do

if ((isPassed[j] = 0) and

((minPath[index] + DijkstraMatrix[index, j]) < minPath[j])) then

minPath[j] := minPath[index] + DijkstraMatrix[index, j]:

path[j] := index:

end if:

end do:

isPassed[index] := 1:

end do:

print("Min path from ", s, " to ", t, " is ", minPath[t]):

print("Path is: "):

j := t:

while (j <> s) do

print ("go to: ", j):

j := path[j]:

end do:

print("end up in: ", j):

end proc:

DijkstraMatrix := copy(A):

print("Dijkstra algorithm: mine"):

for i from 2 to n do

Dijkstra(n, 1, i):

end do:

PinkFloydMatrix := copy(A):

FloydMarshall(n):

print("FloydMarshal algorithm: mine"):

for i from 2 to n do

print("Min path 1 to ", i, " is ", PinkFloydMatrix[1, i]):

end do:

print("Dijkstra algorithm: maple"):

for i from 2 to n do

print("Min path 1 to ", i, " is ", DijkstrasAlgorithm(G, 1, i)):

end do:

GoodBand := AllPairsDistance(G):

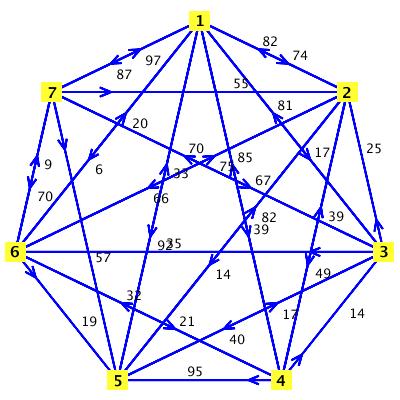
print("FloydMarshall algorithm: maple"):

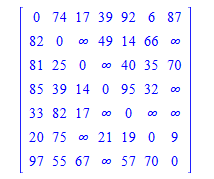
for i from 2 to n do

print("Min path 1 to ", i, "is", GoodBand[1, i]):

end do:

**Результат:**









































































































**Вывод:** в ходе выполнения лабораторной работы составлены программы, реализующие алгоритмы Дейкстры и Флойда-Уоршелла. Изучены возможности пакета GraphTheory для отыскания кратчайших путей на взвешенном графе.

**Список литературы**

1. Белоусов А.И., Ткачёв С.Б. Дискретная математика: Учеб. для вузов / Под ред. В.С. Зарубина, А.П. Крищенко. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2006. – 744 с.
2. Новиков Ф.А. Дискретная математика для бакалавров и магистров. Учебник для вузов. Стандарт третьего поколения. – Спб.: Питер, 2012. – 432 с.