|  |  |
| --- | --- |
|  | ***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 | Турченко С.А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| 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 "data6.txt":

with(GraphTheory):

n := sqrt(ArrayNumElems(A)):

C := copy(A):

for i from 1 to n do

for j from 1 to n do

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

C[i,j] := 0:

end if:

end do:

end do:

G := Graph(C):

DrawGraph(G);

B := copy (A):

weights := AllPairsDistance(G);

FloydWarshall := proc (n)

global B:

local i,j,k:

for k from 1 to n do

for i from 1 to n do

for j from 1 to n do

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

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

end if:

end do:

end do:

end do:

end proc:

Dijkstra := proc (n,start,endDot)

local masResult,masCheck,masIndex,i,j,index,min:

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

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

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

masResult[start] := 0:

for i from 1 to n - 1 do

min := infinity:

for j from 1 to n do

if ((masCheck[j] = 0) and (masResult[j] < min)) then

min := masResult[j]:

index := j:

end if:

end do:

for j from 1 to n do

masResult[index] + A[index,j];

if ((masCheck[j] = 0) and ((masResult[index] + A[index,j]) < masResult[j])) then

masResult[j] := masResult[index] + A[index,j]:

masIndex[j] := index:

end if:

end do:

masCheck[index] := 1:

end do:

j := endDot:

printf ("Weight %a, way: ",masResult[endDot]):

while not(j = start) do

printf(" %a <-",j):

j := masIndex[j]:

end do:

printf(" %a\n",j):

end proc:

printf("\n1)My:\n"):

printf("\nDijkstra:\n"):

for i from 2 to n do

Dijkstra(n,1,i):

end do:

FloydWarshall(n):

printf("\nFloyd-Warshall:\n"):

for i from 2 to n do

printf(" 1 >> %d | %a\n",i,B[1,i]):

end do:

printf("\n2)Maple:\n"):

printf("\nDijkstra:\n"):

for i from 2 to n do

printf(" 1 >> %d | %a\n",i,DijkstrasAlgorithm(G,1,i)):

end do:

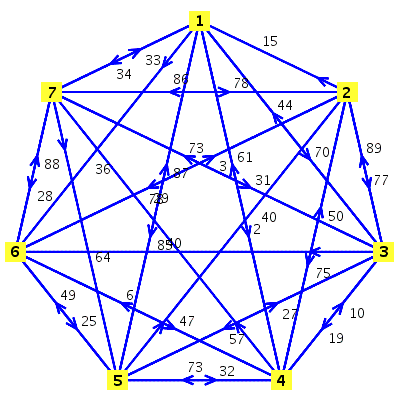
printf("\nFloyd-Warshall:\n"):

for i from 2 to n do

printf(" 1 >> %d | %a\n",i,weights[1,i]):

end do:

**Результат:**





1)My:

Dijkstra:

Weight 11, way: 2 <- 6 <- 4 <- 1

Weight 12, way: 3 <- 4 <- 1

Weight 2, way: 4 <- 1

Weight 33, way: 5 <- 6 <- 4 <- 1

Weight 8, way: 6 <- 4 <- 1

Weight 34, way: 7 <- 1

Floyd-Warshall:

1 >> 2 | 11

1 >> 3 | 12

1 >> 4 | 2

1 >> 5 | 33

1 >> 6 | 8

1 >> 7 | 34

2)Maple:

Dijkstra:

1 >> 2 | [[1, 4, 6, 2], 11]

1 >> 3 | [[1, 4, 3], 12]

1 >> 4 | [[1, 4], 2]

1 >> 5 | [[1, 4, 6, 5], 33]

1 >> 6 | [[1, 4, 6], 8]

1 >> 7 | [[1, 7], 34]

Floyd-Warshall:

1 >> 2 | 11

1 >> 3 | 12

1 >> 4 | 2

1 >> 5 | 33

1 >> 6 | 8

1 >> 7 | 34

**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.