Министерство образования Республики Беларусь

Учреждение образования

«Брестский Государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №3**

По дисциплине «Математические основы интеллектуальных систем»

Тема: «Нахождение кратчайшего пути в графе»

**Выполнил:**

Студент 2 курса

Группы ИИ-21

Карагодин Д.Л.

**Проверил:**

Козинский А.А.

Брест 2022

**Цель работы:** Изучение и решение заданий по нахождению кратчайшего пути в графе.

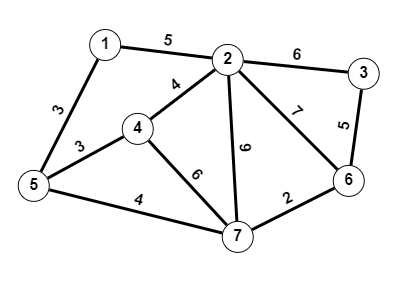
**Ход работы:**

1. Алгоритмом Дейкстры вычислить кратчайшие пути от вершины  ко всем вершинам графа. Варианты графов указаны в таблице 1. Графы заданы списком ребер, в квадратных скобках указаны веса соответствующих ребер.
2. Алгоритмом Флойда-Уоршолла вычислить кратчайшие пути от вершины ко всем вершинам графа.

**Вариант 4**

|  |  |  |  |
| --- | --- | --- | --- |
| №  № | Кол.  вер-  шин | Кол.  ре-  бер | Ребра  и веса |
| 4. | 7 | 11 | {1,2},{1,5},{2,3},{2,4},{2,6}.{2,7},  {3,6},{4,5},{4,7},{5,7},{6,7};  [5,3,6,4,7,9,5,3,6,4,2]; |

**Изображение графа:**



**Код программы:**

#include "1\_lab.hpp"

vector<short> Dijkstra(const VesGraphs& vgh, short st) {

st--;

vector<vector<short>> adj = matrix\_adj\_ves(vgh,false);

vector<short> distance(vgh.max, SHRT\_MAX);

vector<bool> visited(vgh.max,false);

short count=0, index=0, i=0, u=0;

distance[st] = 0;

for (count = 0; count < vgh.max - 1; count++) {

int min = SHRT\_MAX;

for (i = 0; i < vgh.max; i++)

if (!visited[i] && distance[i] <= min) { min = distance[i]; index = i; }

u = index;

visited[u] = true;

for (i = 0; i < vgh.max; i++)

if (!visited[i] && adj[u][i] && distance[u] != SHRT\_MAX &&

distance[u] + adj[u][i] < distance[i])

distance[i] = distance[u] + adj[u][i];

}

for (size\_t i = 0; i < vgh.max;i++ ) cout <<"X\_"<< i+1<<": "<< distance[i] << endl;

return distance;

}

void FU(const VesGraphs& vgh)

{

vector<vector<short>> adj= matrix\_adj\_ves(vgh, false);

int k,i,j;

for (i = 0; i < vgh.max; i++) adj[i][i] = 0;

for (k = 0; k < vgh.max; k++)

for (i = 0; i < vgh.max; i++)

for (j = 0; j < vgh.max; j++)

if (adj[i][k] && adj[k][j] && i != j)

if (adj[i][k] + adj[k][j] < adj[i][j] || adj[i][j] == 0)

adj[i][j] = adj[i][k] + adj[k][j];

for (j = 0; j < vgh.max; j++) cout << adj[0][j] << " ";

}

int main() {

VesGraphs vgh("vesgraph.txt");

vgh.Out();

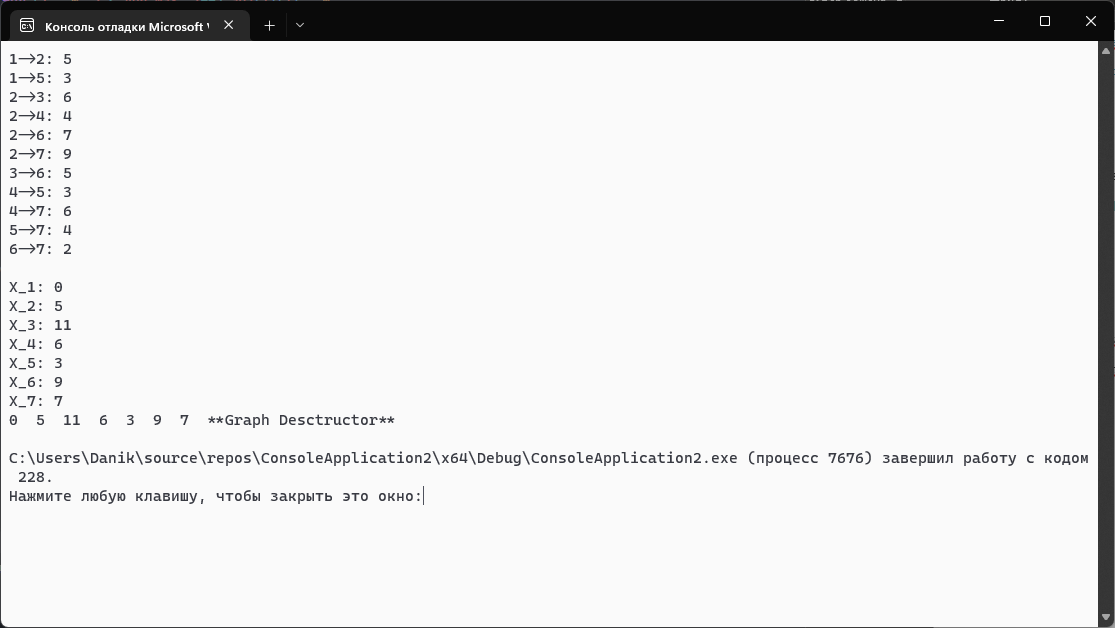
Dijkstra(vgh, 1);

FU(vgh);

return 228;

}

**Результат программы:**



**Вывод:** Изучил и решил задания по нахождению кратчайшего пути в графе.