

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА
ПОЛІТЕХНІКА”**

Кафедра систем штучного інтелекту

Лабораторна робота №5

З дисципліни

“Дискретна математика”

Виконав:

Студент групи КН-115

Конопльов Павло

Викладач:

Мельникова Н.І.

Львів-2019р,

Тема: Знаходження найкоротшого маршруту за алгоритмом Дейкстри. Плоскі планарні графи

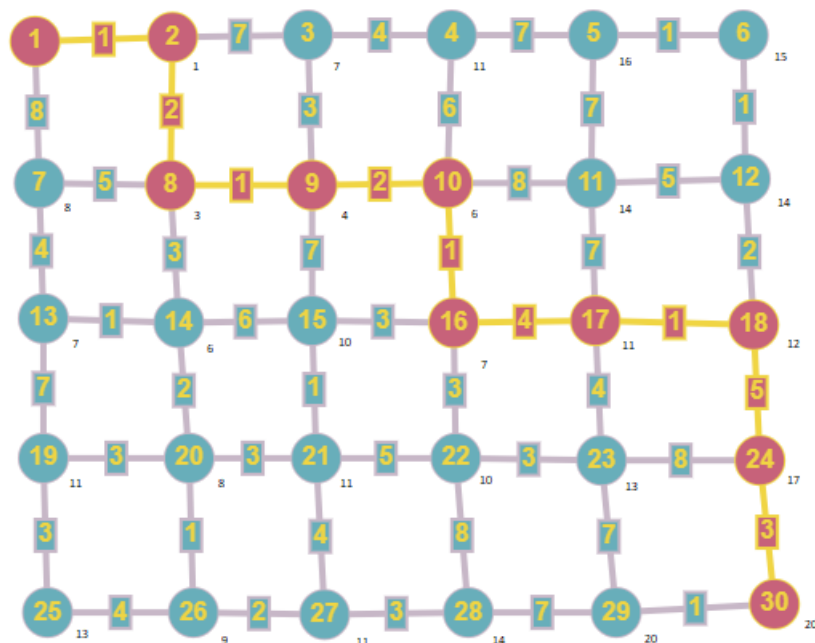
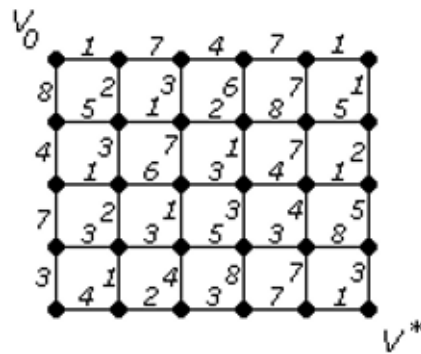
Мета роботи: набуття практичних вмінь та навичок з використання алгоритму Дейкстри.

Варіант № 11

Завдання № 1.

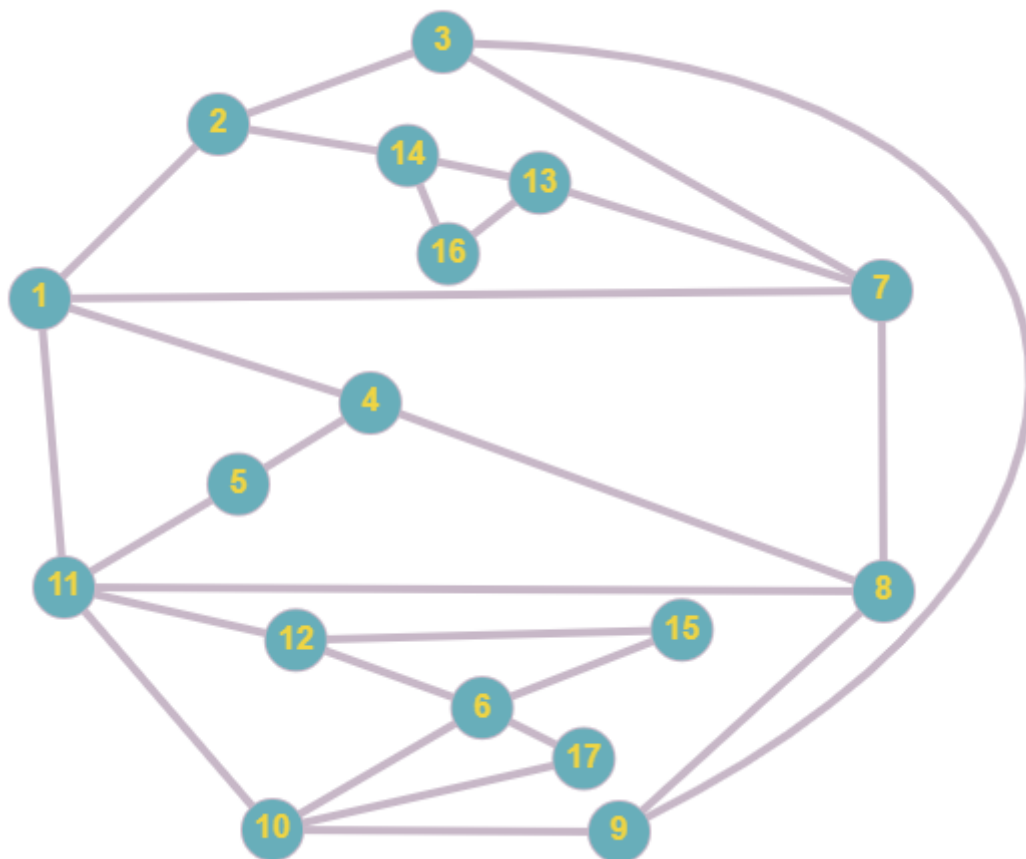
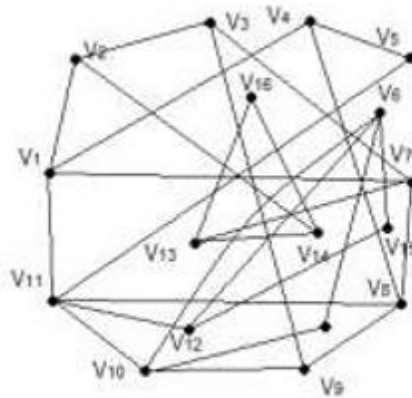
Розв'язати на графах наступні 2 задачі:

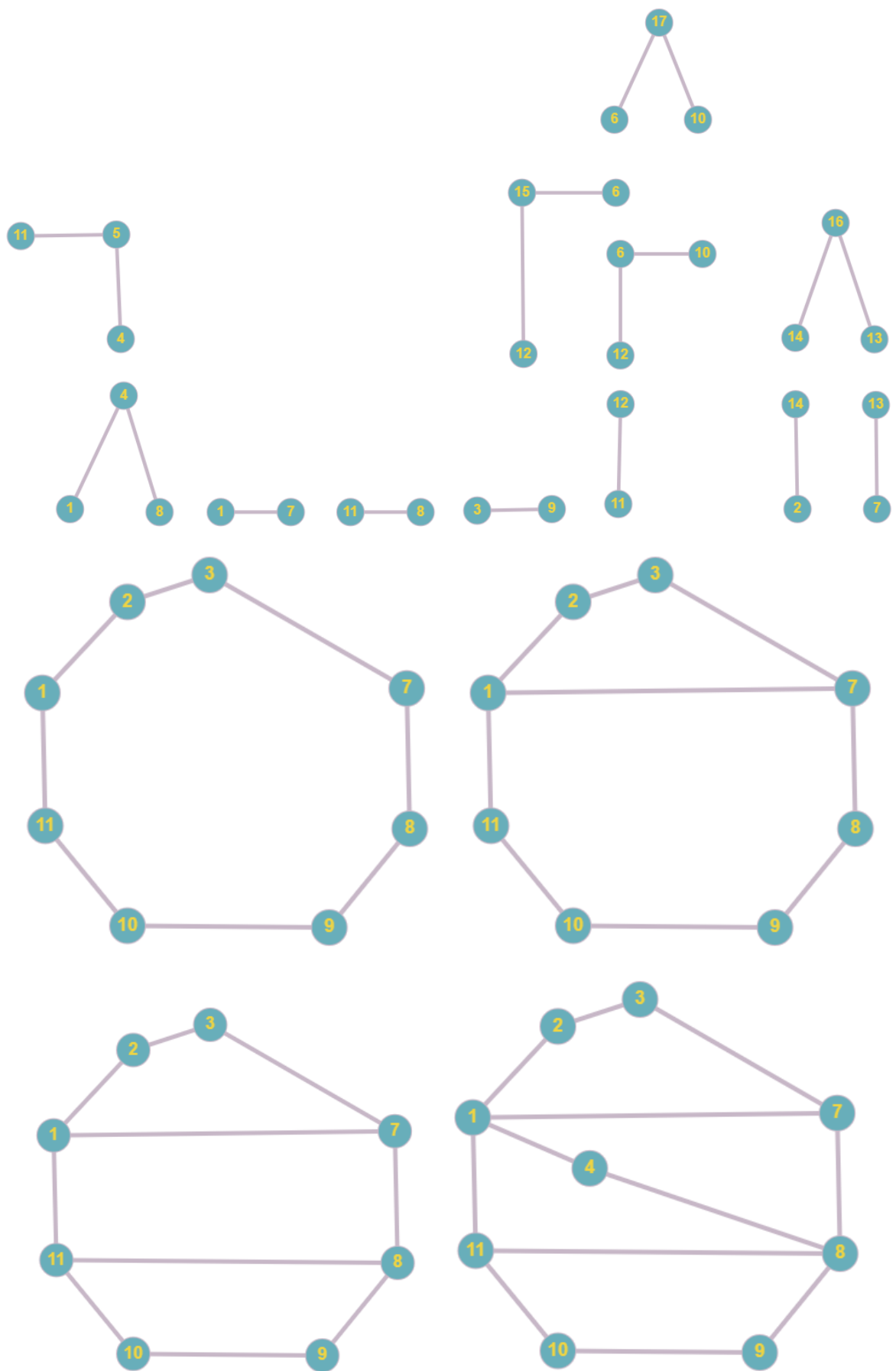
1. За допомогою алгоритму Дейкстри знайти найкоротший шлях у графі поміж парою вершин V_0 і V^* .

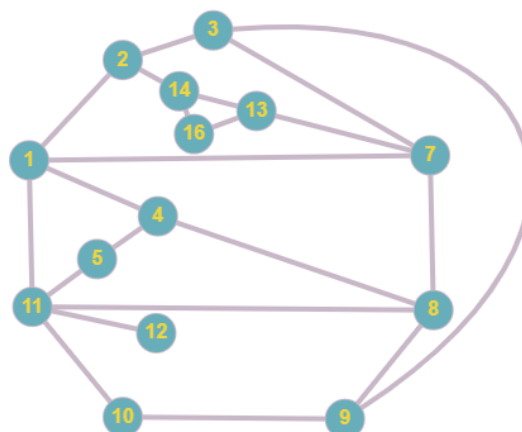
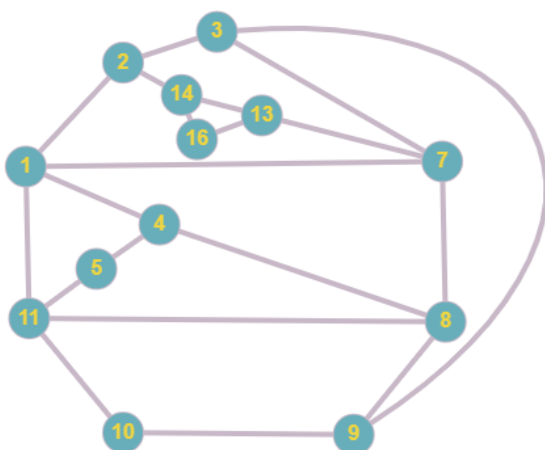
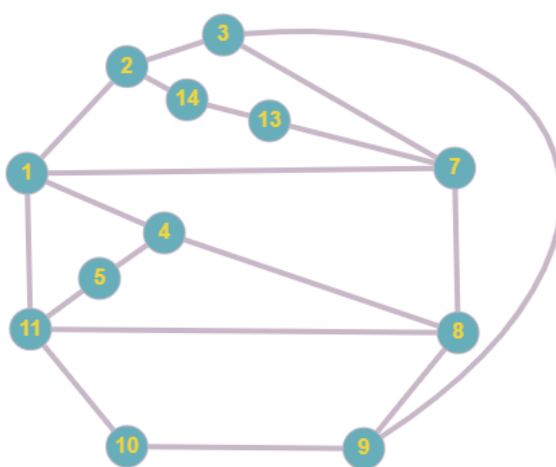
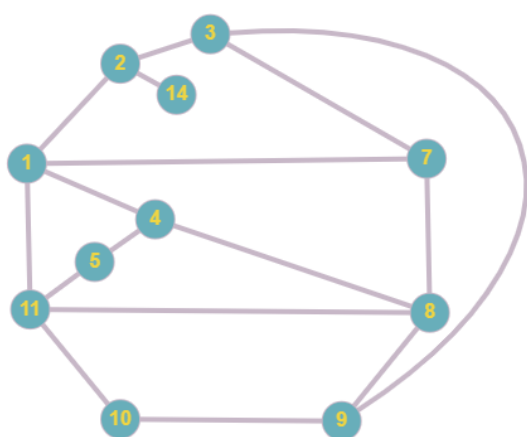
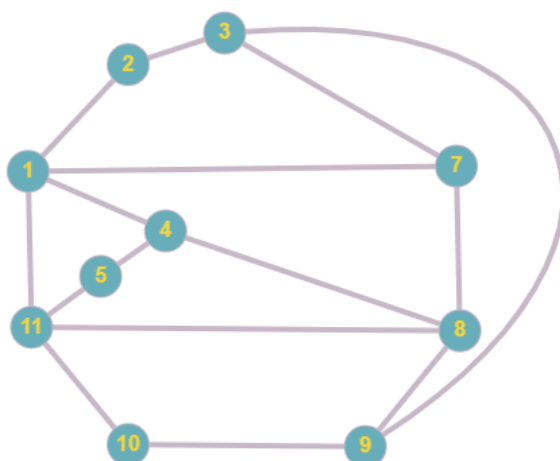
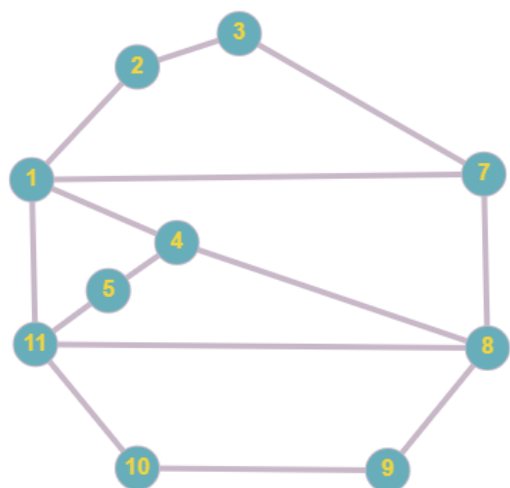


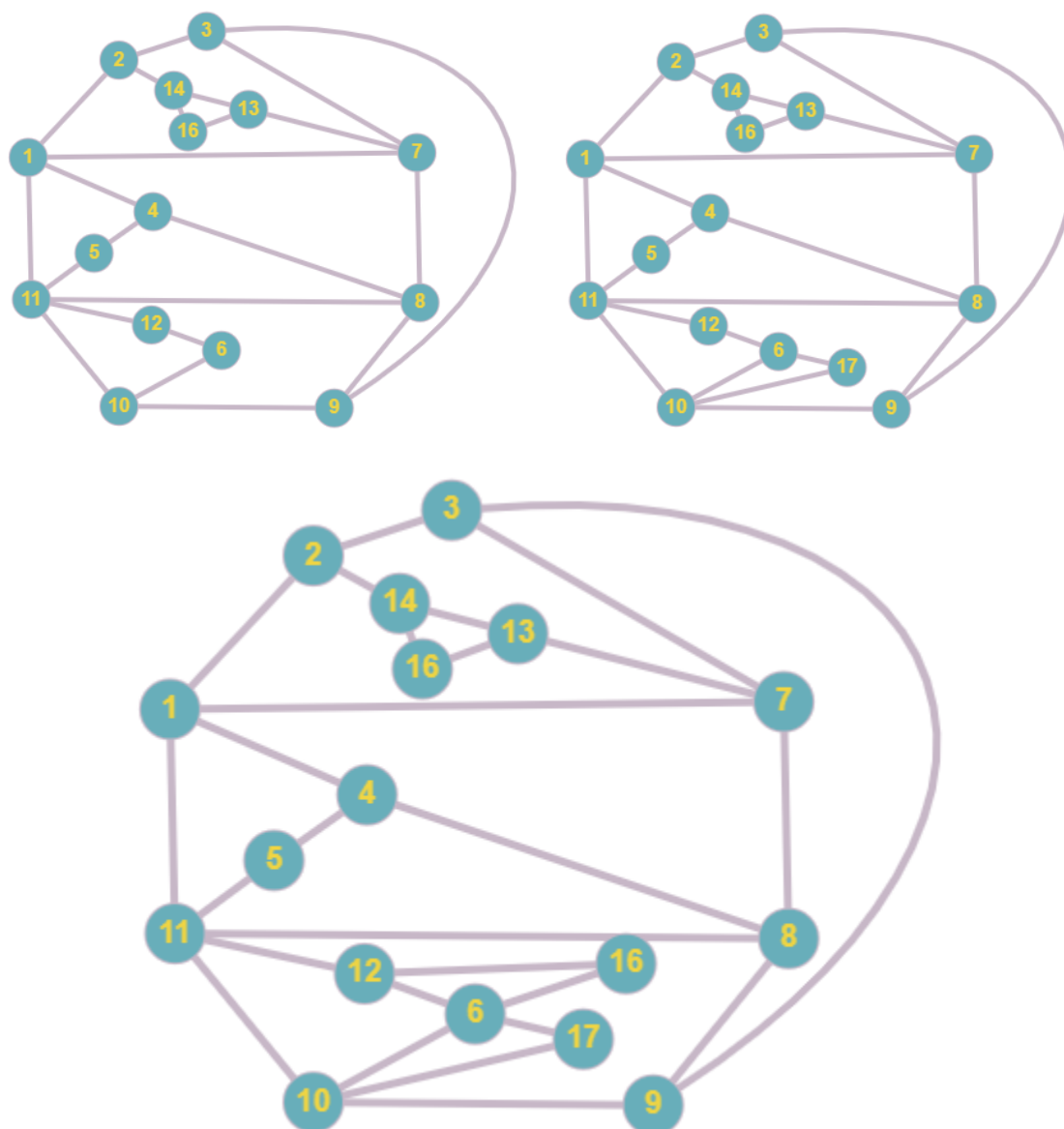
Мінімальна довжина: 20

2. За допомогою γ -алгоритма зробити укладку графа у площині, або довести що вона неможлива.



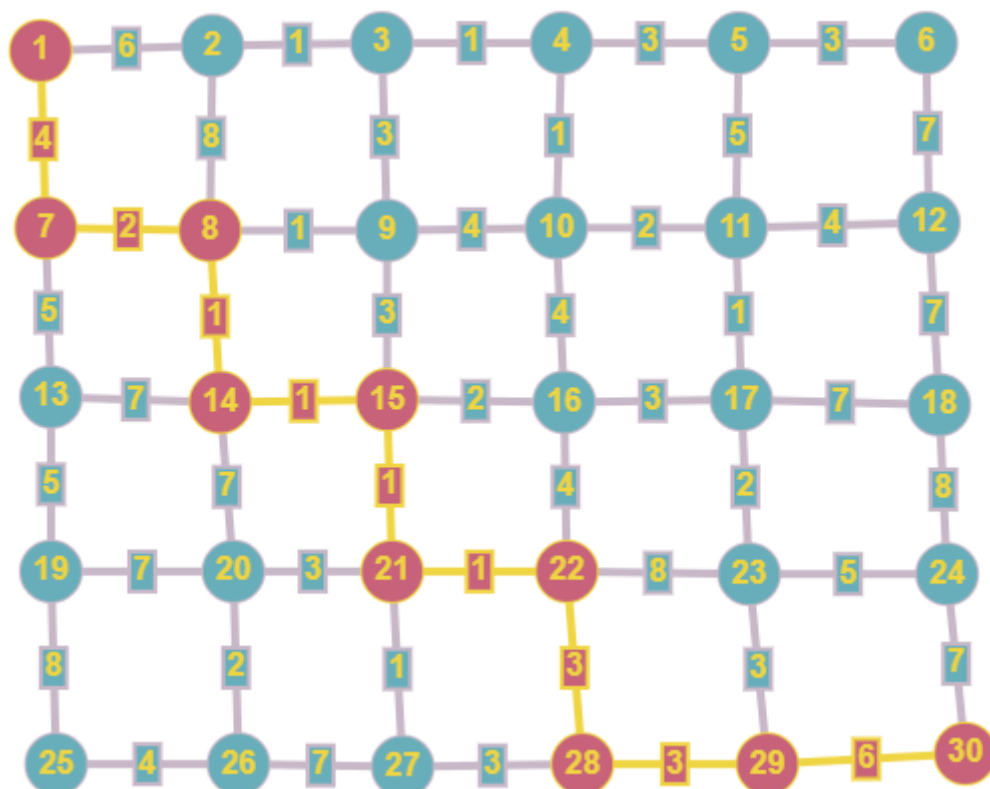
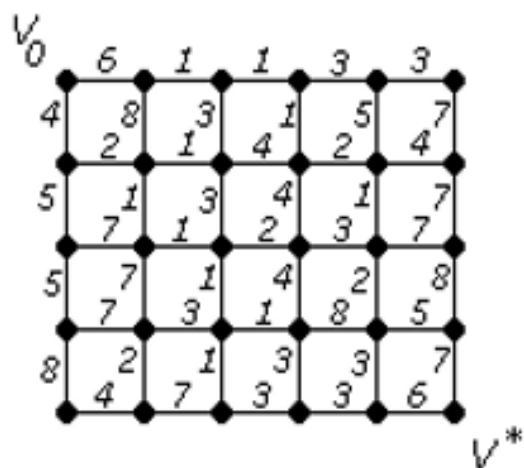






Завдання №2.

Написати програму, яка реалізує алгоритм Дейкстри знаходження найкоротшого шляху між парою вершин у графі. Протестувати розроблену програму на графі згідно свого варіанту.



Матриця суміжності графа:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	0	6	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	6	0	1	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	1	0	1	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	1	0	3	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	3	0	3	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	3	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	4	0	0	0	0	0	0	2	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	8	0	0	0	0	2	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	3	0	0	0	0	1	0	4	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	1	0	0	0	0	4	0	2	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0
11	1	0	0	0	0	5	0	0	0	0	2	0	4	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	7	0	0	0	0	4	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	5	0	0	0	0	0	0	7	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	1	0	0	0	0	7	0	1	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	3	0	0	0	0	1	0	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	4	0	0	0	0	2	0	3	0	0	0	0	4	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	3	0	7	0	0	0	0	2	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	7	0	0	0	0	0	8	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	7	0	0	0	8	0	0	0	0	0	0

[illegible]

Приклад реалізації програми:

```
#include <iostream>
#include<iomanip>

using namespace std;

int main()
{
    int a = 1000;
    int Mtx[100][100];
    int x[100];
    int LengthWay[100];
    int PreVertex[100];
    int size;

    cout << "Amount of verticles: ";
    cin >> size;
    cout << "Enter weight matrix: " << endl;

    for (int i = 0; i < size; i++)
    {
        for (int j = 0; j < size; j++)
        {
            cin >> Mtx[i][j];
        }
    }

    int start;
    int end;

    do
    {
        cout << "Enter start vertex: ";
        cin >> start;
    } while (start > (size - 1) || start < 0);

    int v;

    start = start - 1;

    for (int i = 0, j = 0; i < size; i++, j++)
    {
        end = i;
        if (start == end) continue;
        else
        {
            int u;
            for (u = 0; u < size; u++)
```

```

int u;
for (u = 0; u < size; u++)
{
    LengthWay[u] = a;
    x[u] = 0;
}

PreVertex[start] = 0;
LengthWay[start] = 0;
x[start] = 1;
v = start;

while (1)
{
    for (u = 0; u < size; u++)
    {
        if (Mtx[v][u] == 0) continue;

        if (x[u] == 0 && LengthWay[u] > LengthWay[v] + Mtx[v][u])
        {
            LengthWay[u] = LengthWay[v] + Mtx[v][u];
            PreVertex[u] = v;
        }
    }

    int w = a;

    for (u = 0; u < size; u++)
    {
        if (x[u] == 0 && LengthWay[u] < w)
        {
            v = u;
            w = LengthWay[u];
        }
    }

    if (v == end) break;

    x[v] = 1;
}

int fin;

```

```

int fin;

do
{
    cout << "Enter finish vertex: ";
    cin >> fin;
} while (fin > (size+1) || fin < 0);

end = fin-1;
int u = fin-1;

if (fin == end + 1)
{
    cout << "Smallest way from " << start + 1 << " in vertex " << end + 1 << " :";
    int u = end;

    while (u != start)
    {
        cout << " " << u + 1;
        u = PreVertex[u];
    }
    cout << " " << start + 1 << " And cost --> " << LengthWay[end];
}

```

Результат виконання програми:

```
Amount of vertices: 30
Enter weight matrix:
0 6 0 0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
6 0 1 0 0 0 0 8 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 0 1 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 0 3 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 3 0 3 0 0 0 0 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 3 0 0 0 0 0 0 7 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
4 0 0 0 0 0 0 2 0 0 0 0 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 8 0 0 0 0 2 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 3 0 0 0 0 1 0 4 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 4 0 2 0 0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 5 0 0 0 0 2 0 4 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 7 0 0 0 0 4 0 0 0 0 0 0 7 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 5 0 0 0 0 0 0 7 0 0 0 0 5 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0 0 7 0 1 0 0 0 0 7 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 3 0 0 0 0 1 0 2 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 4 0 0 0 0 2 0 3 0 0 0 0 4 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 3 0 7 0 0 0 0 2 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 7 0 0 0 0 0 7 0 0 0 0 8 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 5 0 0 0 0 0 7 0 0 0 8 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 7 0 0 0 0 7 0 3 0 0 0 0 2 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 3 0 1 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 4 0 0 0 0 1 0 8 0 0 0 0 3 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 8 0 5 0 0 0 0 3 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 8 0 0 0 0 5 0 0 0 0 0 7 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 8 0 0 0 0 0 0 4 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 4 0 7 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 7 0 3 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 3 0 3 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 3 0 6 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 7 0 0 0 0 6 0 0 0 0 0
Enter start vertex: 1
Enter finish vertex: 30
Smallest way from 1 in vertex 30 : 30 29 28 22 21 15 14 8 7 1
And cost --> 22
```

Висновок: під час виконання лабораторної роботи я навчився використовуючи алгоритм Дейкстри для знаходження найкоротшого маршруту. Також дізнався алгоритм укладання графа на площині.