

**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

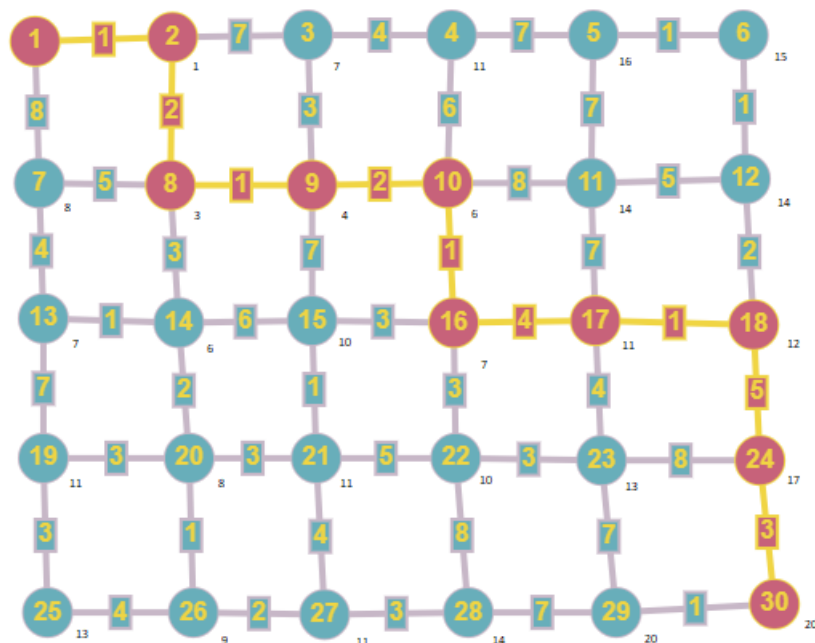
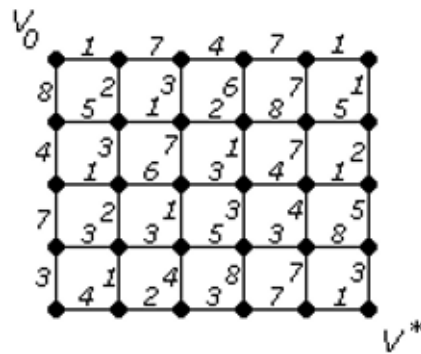
**Мета роботи:** набуття практичних вмінь та навичок з використання алгоритму Дейкстри.

## Варіант № 11

### Завдання № 1.

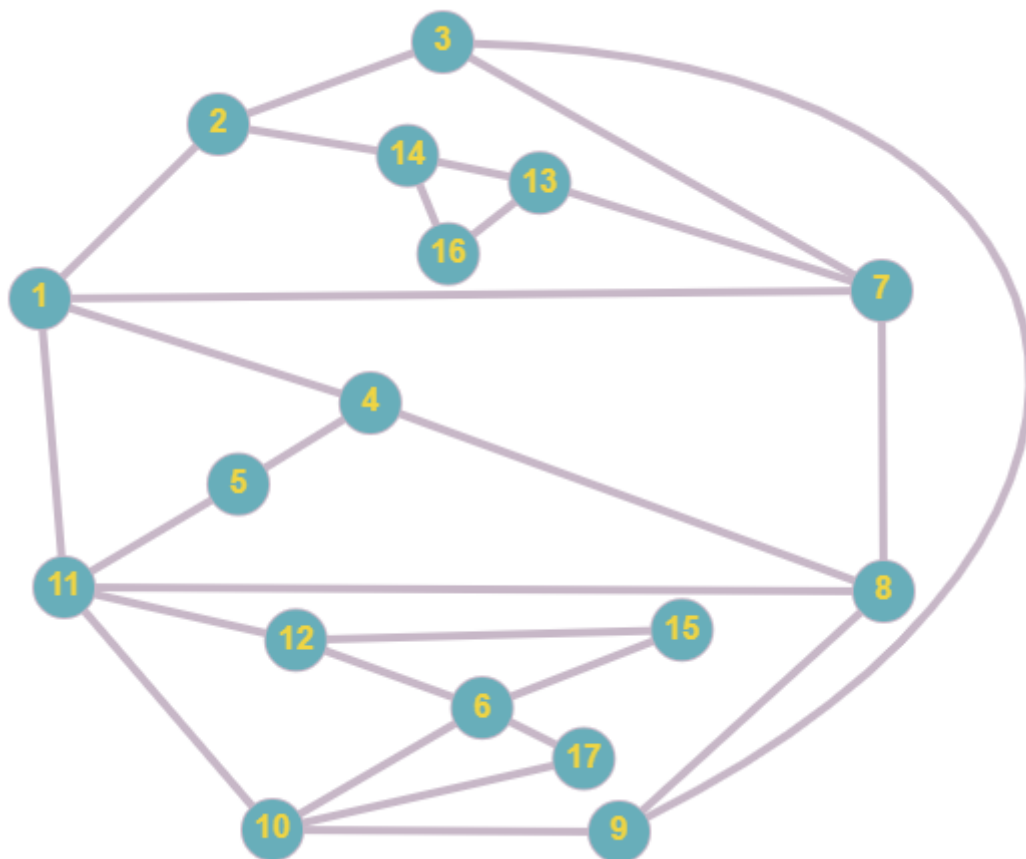
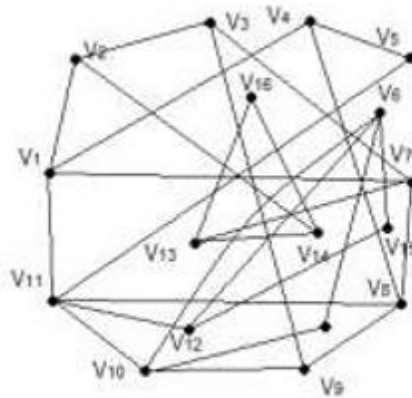
Розв'язати на графах наступні 2 задачі:

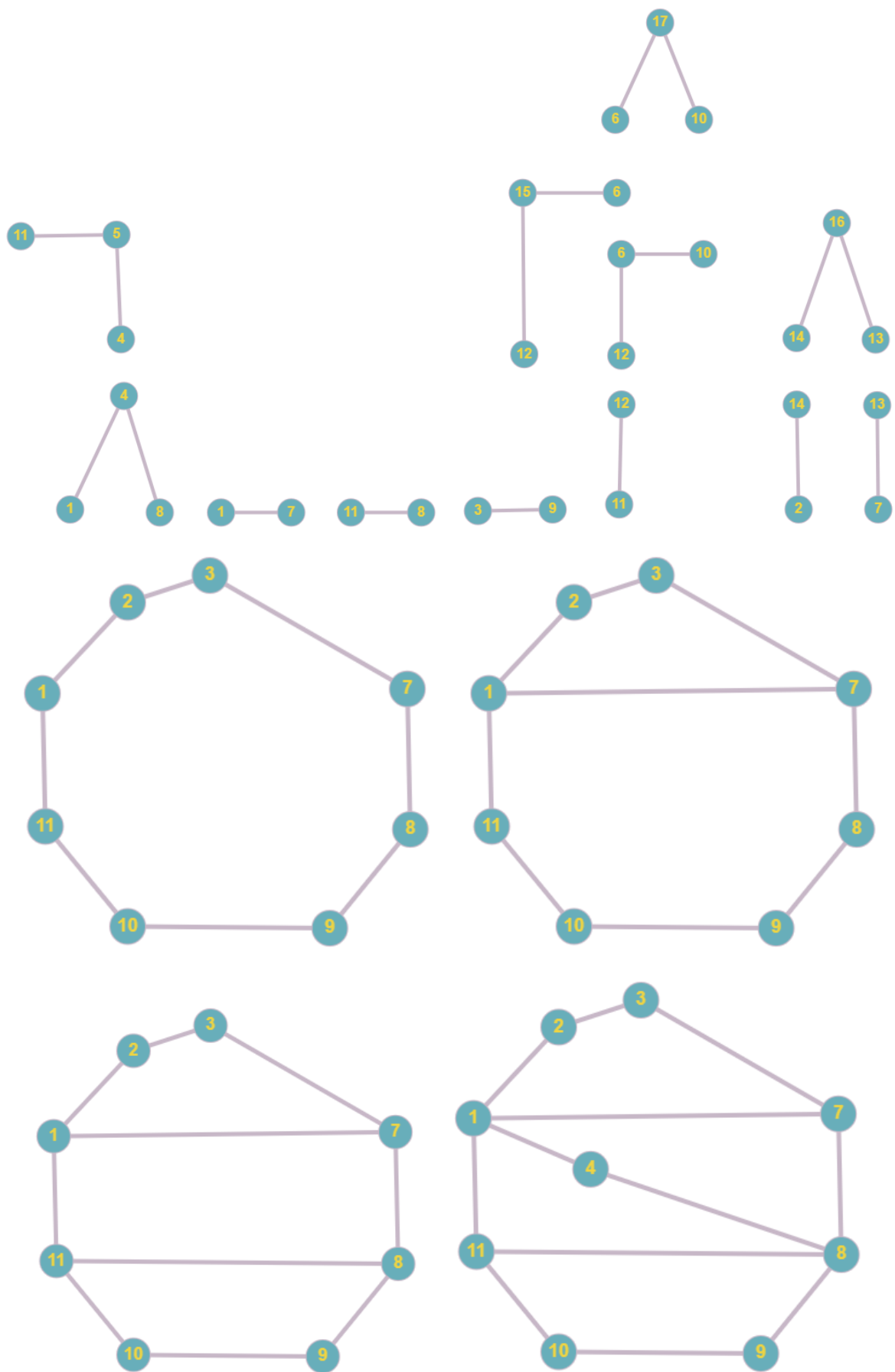
1. За допомогою алгоритму Дейкстри знайти найкоротший шлях у графі поміж парою вершин  $V_0$  і  $V^*$ .

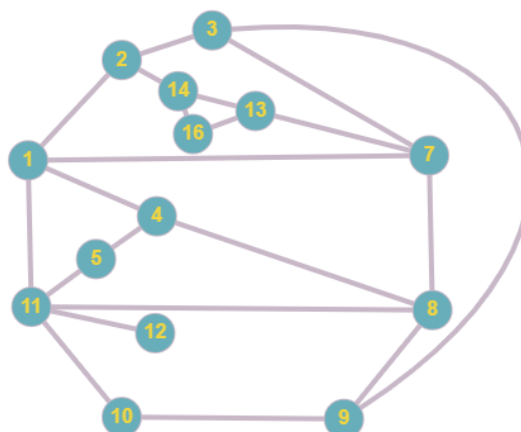
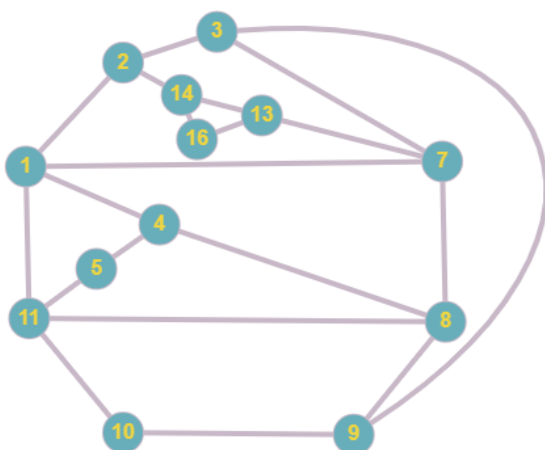
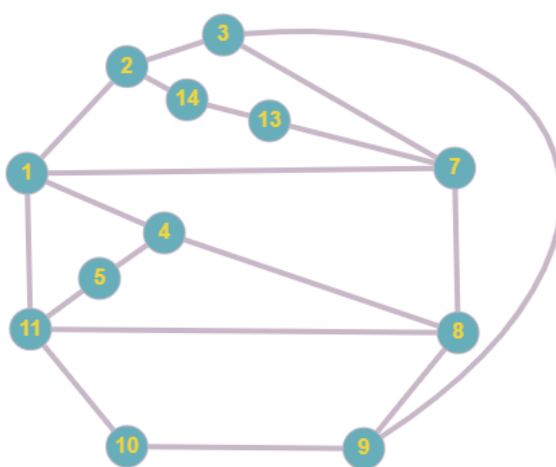
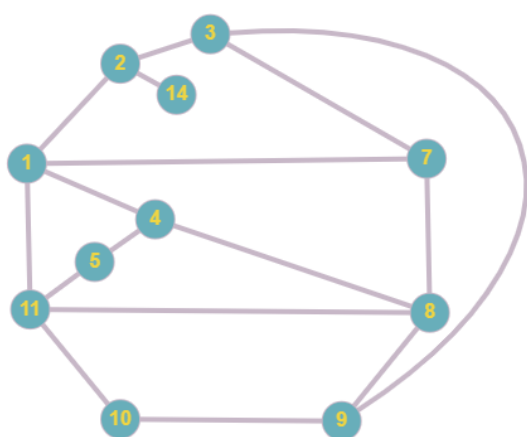
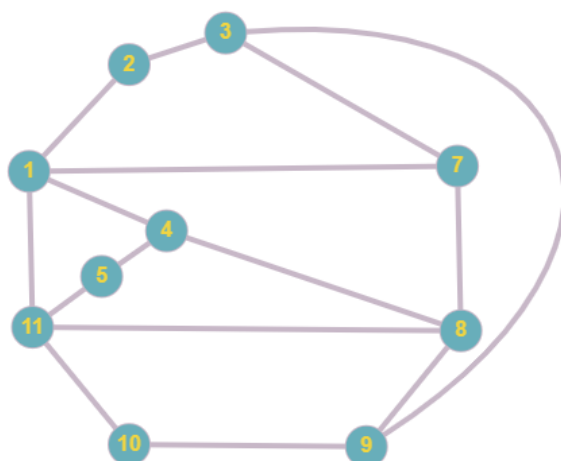
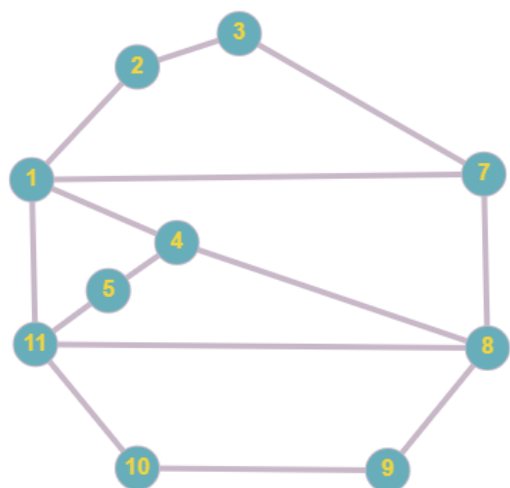


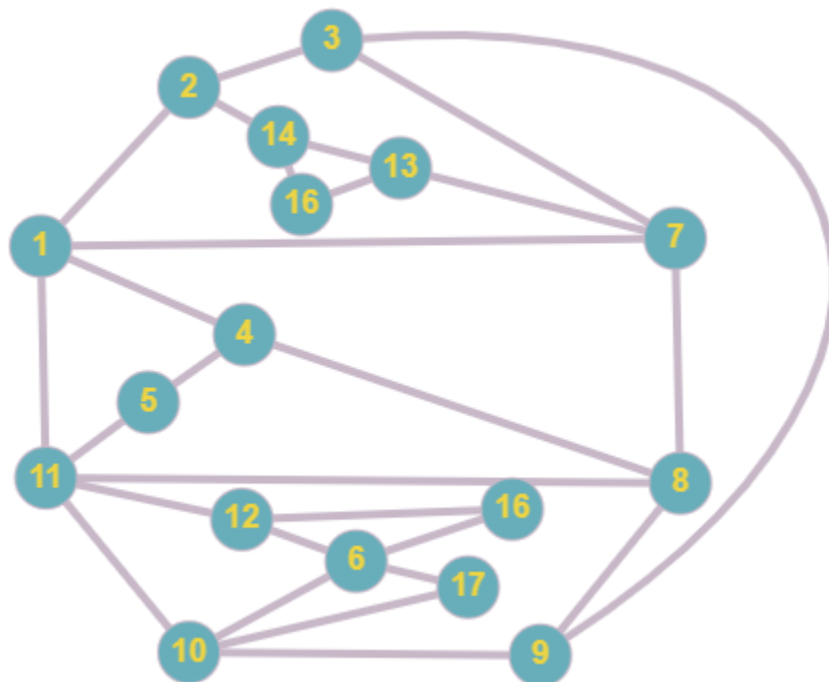
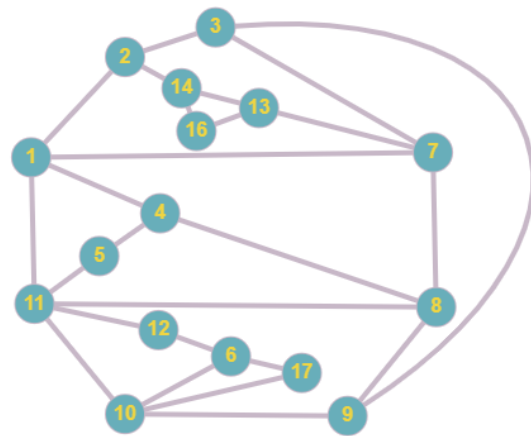
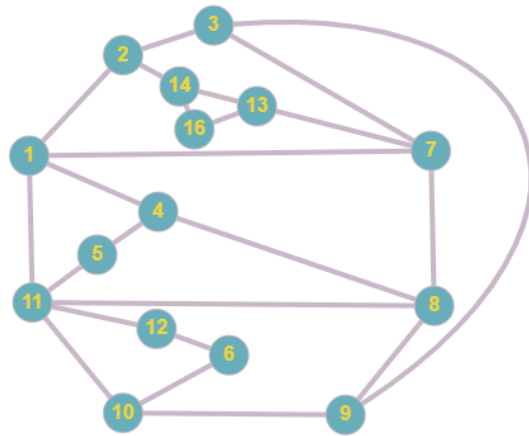
Мінімальна довжина: 20

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



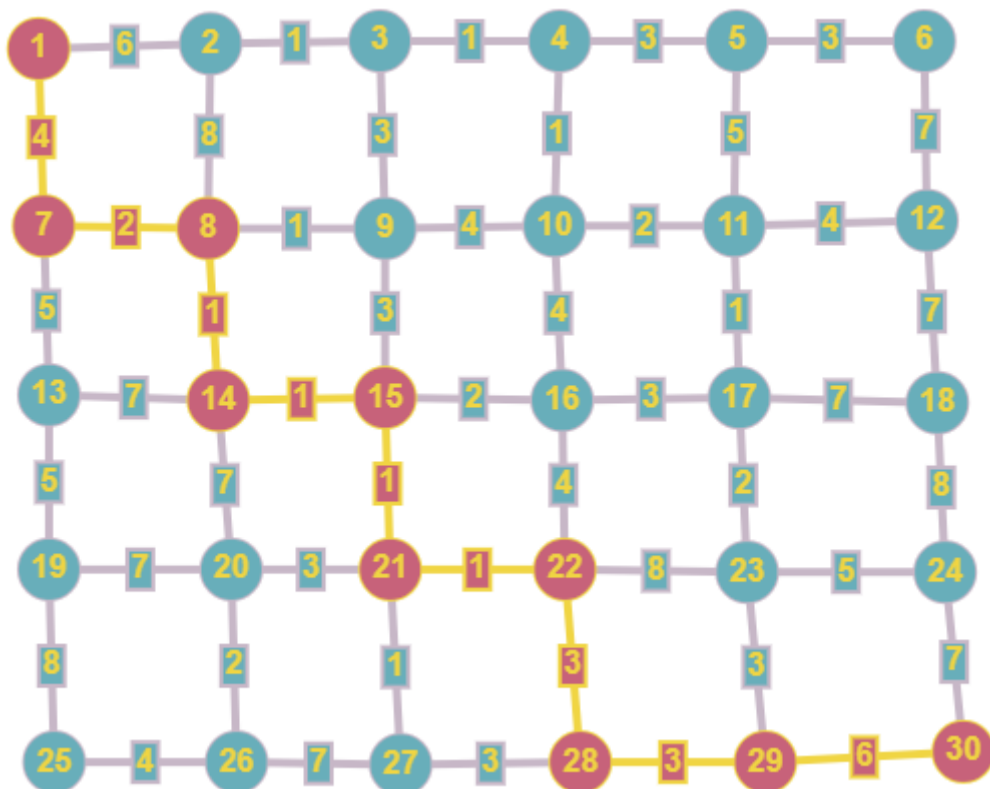
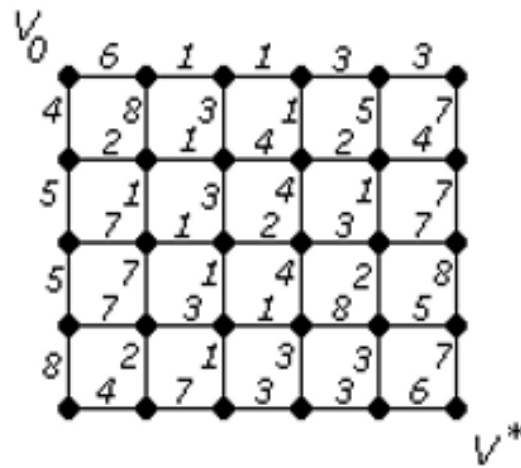






## Завдання №2.

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	1	0	0	0	0	3	0	7	0	0	0	0	2	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	7	0	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	0	8	0	0	0	0	0

20	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
21	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
22	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
23	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
24	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	0	0	7	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	4	0	0	0	0	0
26	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
27	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
28	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	3	0	3	0	0	0
29	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
03	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

### Приклад реалізації програми:

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

using namespace std;

int main()
{
    int i, j;
    int infinity = 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;
    cout << setw(4);
    for (i = 0; i < size; i++)
    {
        cout << "|V" << i + 1;
    }
    cout << endl;

    for (i = 0; i < size; i++)
    {
        cout << "V" << i + 1 << '|';
        for (j = 0; j < size; j++)
        {
            cin >> Mtx[i][j];
        }
    }

    cout << " " << endl;

    int start;
    int end;
```

```

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

cout << " " << endl;
cout << "---Smallest way list---" << endl;
cout << " " << endl;

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++)
        {
            LengthWay[u] = infinity;
            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];

```

```

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

    int w = infinity;
    v = -1;

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

    if (v == end)
    {
        cout << "[" << j << "]" << " Smallest way from " << start + 1 << " in vertex " << end + 1 << " :";
        u = end;

        while (u != start)
        {
            cout << " " << u + 1;
            u = PreVertex[u];
        }
        cout << " " << start + 1 << ". Way - " << LengthWay[end];
        cout << endl;
        break;
    }
    x[v] = 1;
}
}
}

```

## Результат виконання програми:

```
Amount of vertices: 30
Enter weight matrix:
  |v1|v2|v3|v4|v5|v6|v7|v8|v9|v10|v11|v12|v13|v14|v15|v16|v17|v18|v19|v20|v21|v22|v23|v24|v25|v26|v27|v28|v29|v30
v1| 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
v2| 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
v3| 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
v4| 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
v5| 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
v6| 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
v7| 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
v8| 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
v9| 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
v10| 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
v11| 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
v12| 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
v13| 0  0  0  0  0  0  5  0  0  0  0  0  0  0  0  0  7  0  0  0  0  0  0  0  0  0  0  0  0  0
v14| 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
v15| 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
v16| 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
v17| 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
v18| 0  0  0  0  0  0  0  0  0  0  0  7  0  0  0  0  0  7  0  0  0  0  0  8  0  0  0  0  0  0
v19| 0  0  0  0  0  0  0  0  0  0  0  0  5  0  0  0  0  0  0  7  0  0  0  0  8  0  0  0  0  0
v20| 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
v21| 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
v22| 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
v23| 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  3
v24| 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
v25| 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
v26| 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
v27| 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
v28| 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
v29| 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
v30| 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

Enter start vertex: 1

---Smallest way list---

[1] Smallest way from 1 in vertex 2 : 2 1. Way - 6
[2] Smallest way from 1 in vertex 3 : 3 2 1. Way - 7
[3] Smallest way from 1 in vertex 4 : 4 3 2 1. Way - 8
[4] Smallest way from 1 in vertex 5 : 5 4 3 2 1. Way - 11
[5] Smallest way from 1 in vertex 6 : 6 5 4 3 2 1. Way - 14
[6] Smallest way from 1 in vertex 7 : 7 1. Way - 4
[7] Smallest way from 1 in vertex 8 : 8 7 1. Way - 6
[8] Smallest way from 1 in vertex 9 : 9 8 7 1. Way - 7
[9] Smallest way from 1 in vertex 10 : 10 4 3 2 1. Way - 9
[10] Smallest way from 1 in vertex 11 : 11 10 4 3 2 1. Way - 11
[11] Smallest way from 1 in vertex 12 : 12 11 10 4 3 2 1. Way - 15
[12] Smallest way from 1 in vertex 13 : 13 7 1. Way - 9
```

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