

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

**Кафедра систем штучного інтелекту**

## **Лабораторна робота №5**

**з дисципліни «Дискретна математика»**

**Виконав:**

студент групи КН-112

Тиський Святослав

**Викладач:**

Мельникова Н.І.

## Варіант №14

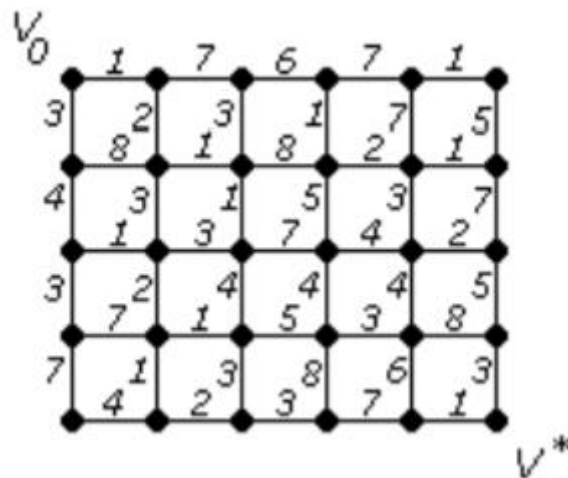
### Лабораторна робота № 5.

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

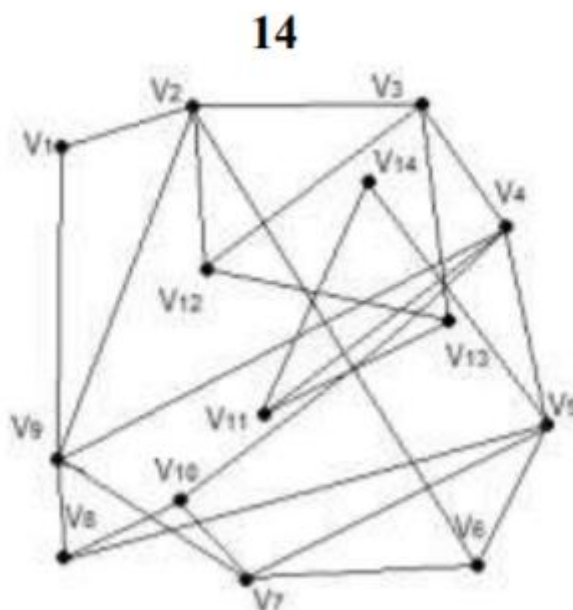
**Мета роботи:** набуття практичних вмінь та навичок з використання алгоритму Дейкстри.

**Завдання № 1:** Розв'язати на графах наступні 2 задачі:

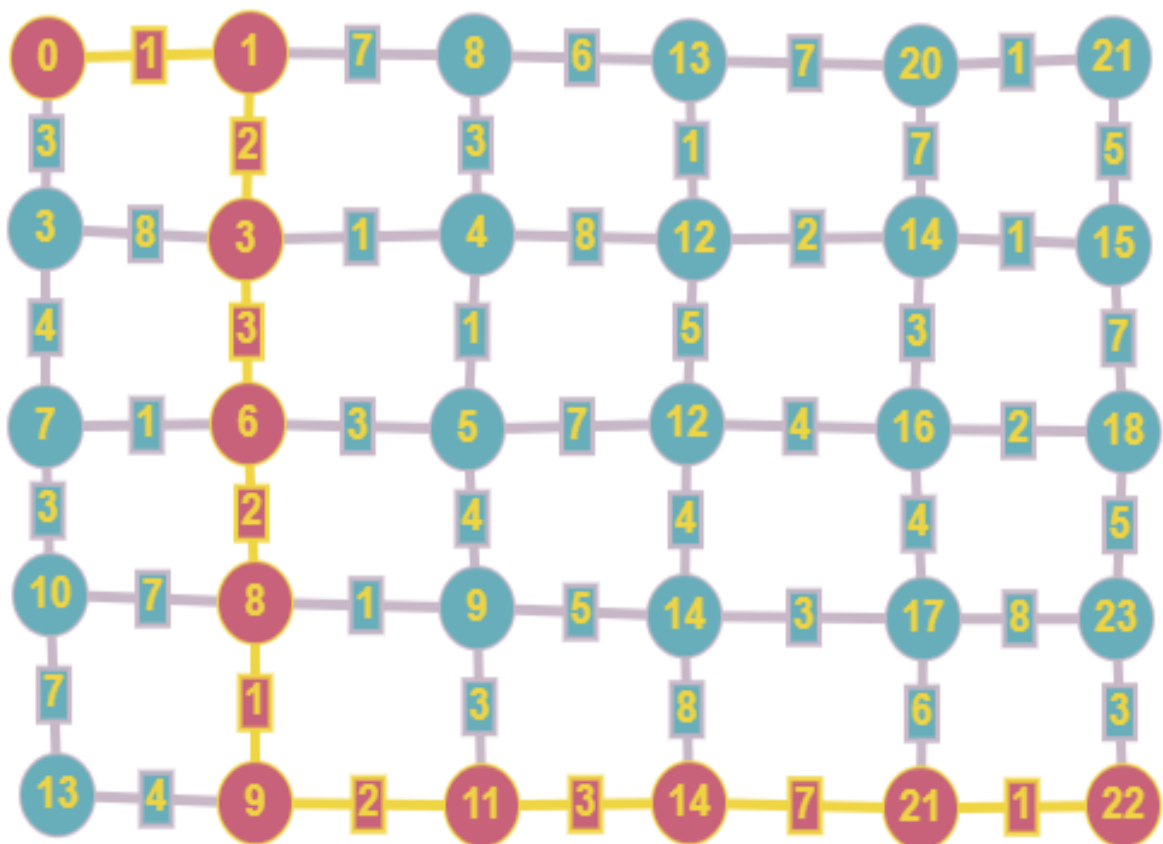
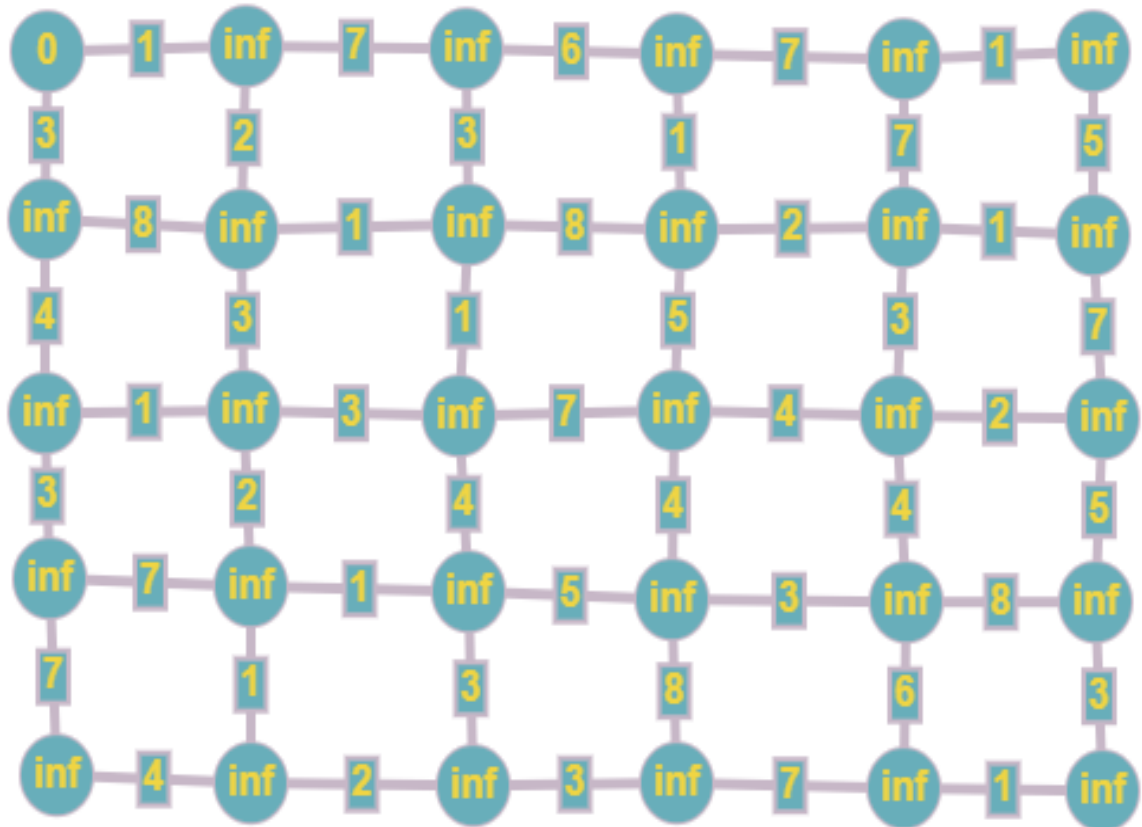
1. За допомогою алгоритму Дейкстри знайти найкоротший шлях у графі поміж парою вершин  $V_0$  і  $V^*$ .



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

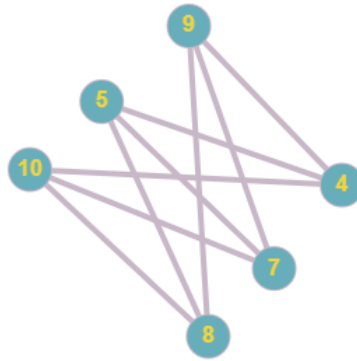


# Розв'язок завдання №1.1:

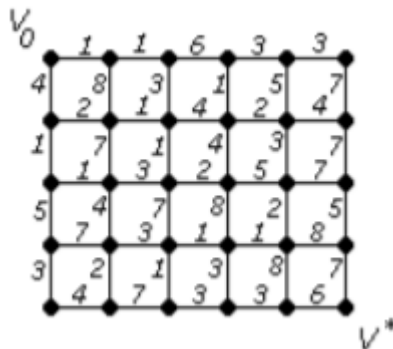


## Розв'язок завдання №1.2:

Перед тим як починати укласти граф слід перевірити, чи можливо цей граф укласти. Перевіримо чи граф є планарний. У цьому графі можна виділити повний дводольний підграф, що означає що цей граф є непланарний, отже укласти його не можна.



**Завдання №2:** Написати програму, яка реалізує алгоритм Дейкстри знаходження найкоротшого шляху між парою вершин у графі. Протестувати розроблену програму на графі згідно свого варіанту.



## Розв'язок завдання №2:

```
1 #include <iostream>
2 #define inf 9999
3 using namespace std;
4 int min_top(int** arr, int v) {
5     int m = 0;
6     for (int i = 0; i < v; i++) {
7         if (arr[i][1]) {
8             m = i; break;
9         }
10    }
11
12    for (int i = 1; i < v; i++) {
13        if (arr[m][0] >= arr[i][0] && arr[i][1] == 1) {
14            m = i;
15        }
16    }
17    return m;
18 }
19
20 int main()
21 {
22     setlocale(LC_ALL, "Ukrainian");
23
24     int a, b, c;
25     int v = 0;
26     cout << "Кількість вершин графа : ";
27     cin >> v;
28     int** graph = new int* [v];
29
30     for (int j = 0; j < v; j++) {
31         graph[j] = new int[v];
32     }
33     cout << "Введіть матрицю суміжності: "<<endl;
34     for (int a = 0; a < v; a++) {
35         for (int j = 0; j < v; j++) {
36             cin >> graph[a][j];
37         }
38     }
39
40     int p;
41     int** tops = new int* [v];
42     for (int j = 0; j < v; j++) {
43         tops[j] = new int[2];
```

```

43     tops[j] = new int[2];
44 }
45 int* tops_path = new int[v];
46
47 cout << "Вихідна вершина: ";
48 cin >> p;
49
50 for (int i = 0; i < v; i++) {
51     if (i == p - 1) {
52         tops[i][0] = 0;
53         tops[i][1] = 1;
54     }
55     else {
56         tops[i][0] = inf;
57         tops[i][1] = 1;
58     }
59 }
60 tops_path[p - 1] = 0;
61
62
63
64 int m;
65
66 for (int i = 0; i < v; i++) {
67     m = min_top(tops, v);
68     for (int j = 0; j < v; j++) {
69         if (graph[m][j]) {
70             if (tops[j][0] > tops[m][0] + (graph[m][j])) {
71                 tops[j][0] = tops[m][0] + (graph[m][j]);
72                 tops_path[j] = m;
73             }
74         }
75     }
76     tops[m][1] = 0;
77 }
78
79
80
81 ///шлях
82 cout << "Введіть потрібну вершину: ";
83 int k; cin >> k;
84 cout << "Мінімальний шлях: ";
85 cout << tops[k - 1][0];
86 cout << endl << k << " <-- ";
87 k--;
88 for (int a = 0; tops_path[k] != p - 1; a++) {
89     cout << tops_path[k] + 1 << " <-- ";
90     k = tops_path[k];
91 }
92 cout << p << endl;
93 return 0;
94 }

```

```

Кількість вершин графа : 30
Введіть матрицю суміжності:
0 1 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 0
1 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
0 1 0 6 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 0 6 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 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 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 0
4 0 0 0 0 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 0
0 8 0 0 0 0 2 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 0 3 0 0 0 0 1 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 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 0 5 0 0 0 0 2 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 0 7 0 0 0 0 4 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 0 1 0 0 0 0 0 1 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 7 0 0 0 0 1 0 3 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 1 0 0 0 0 3 0 2 0 0 0 7 0 0 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 5 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 3 0 0 0 0 5 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 0 7 0 0 0 0 7 0 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 5 0 0 0 0 0 7 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 4 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 7 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 0 8 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 2 0 0 0 0 1 0 8 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 8 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 0 3 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 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 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 0 8 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 0 0 7 0 0 0 0 6 0 0 0 0 0 0 0
Вихідна вершина: 1
Введіть потрібну вершину: 30
Мінімальний шлях: 26
30 <-- 29 <-- 28 <-- 27 <-- 21 <-- 15 <-- 9 <-- 3 <-- 2 <-- 1

```

**Висновок:** на цій лабораторній роботі я набув практичних вмінь та навичок з використання алгоритму Дейкстри.