

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

**Кафедра систем штучного інтелекту**

**Лабораторна робота № 5**

з дисципліни

«Дискретна математика»

**Виконав:**

студент групи КН-112

Стаськів Максим

**Викладач:**

Мельникова Н.І.

Львів – 2019р.

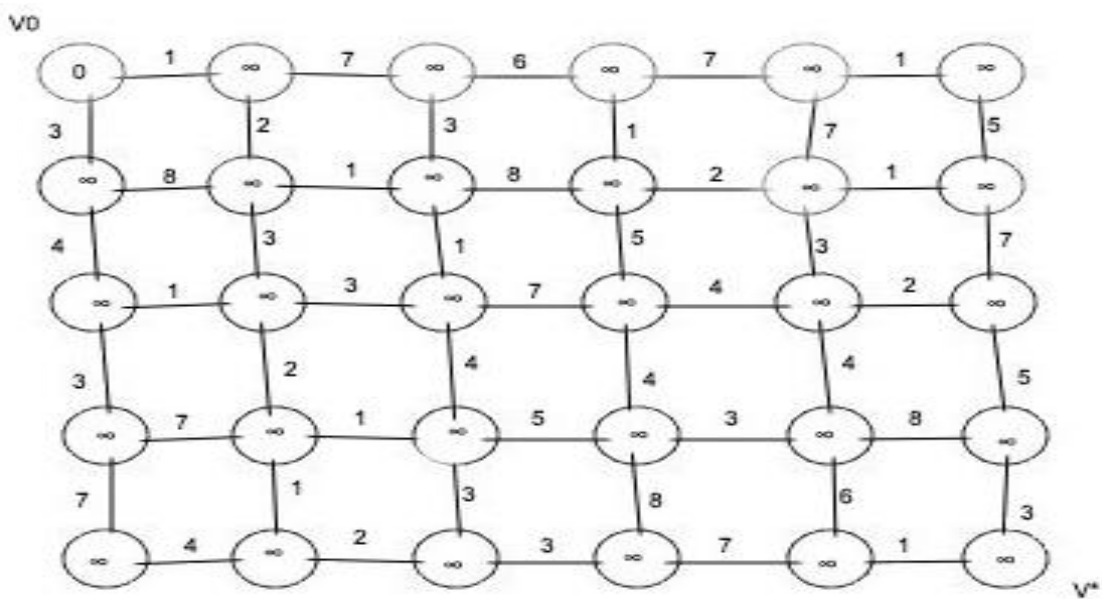
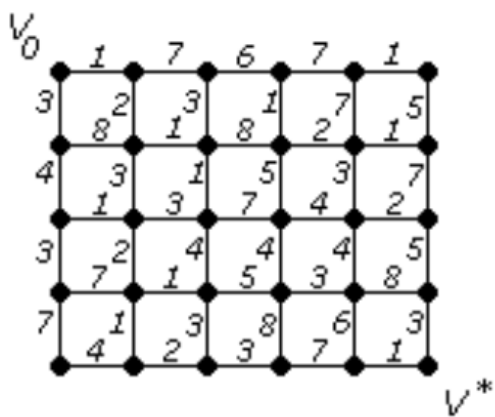
**Варіант 14**

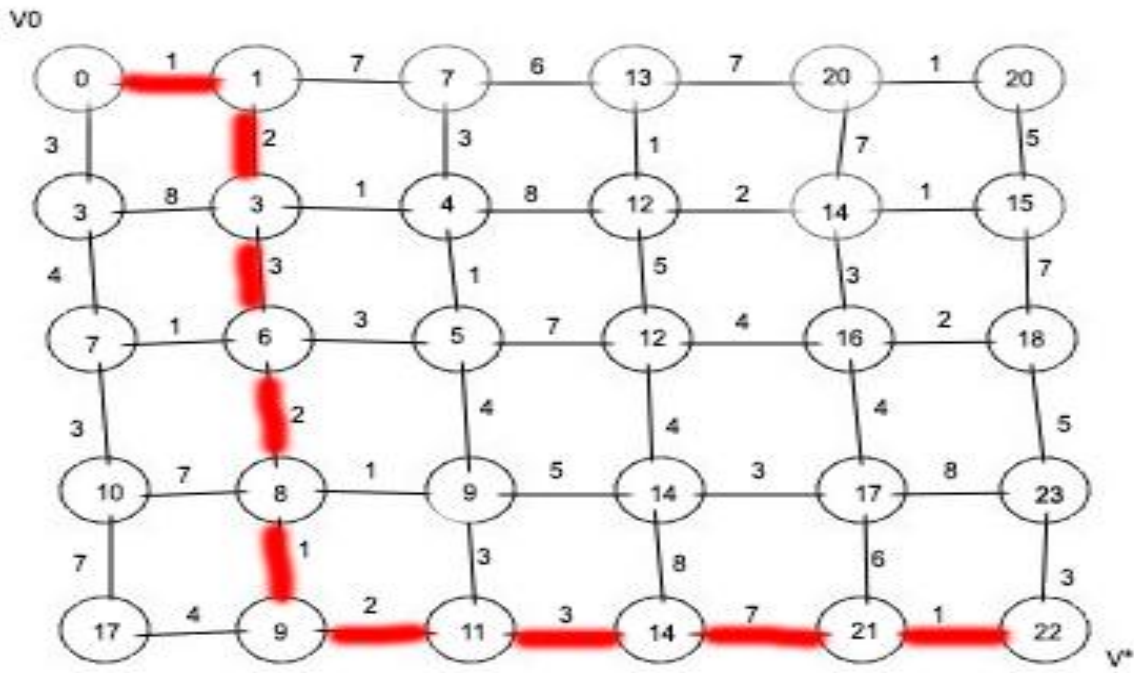
**Тема:** Знаходження найкоротшого маршруту за алгоритмом Дейкстри. Плоскі планарні графи

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

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

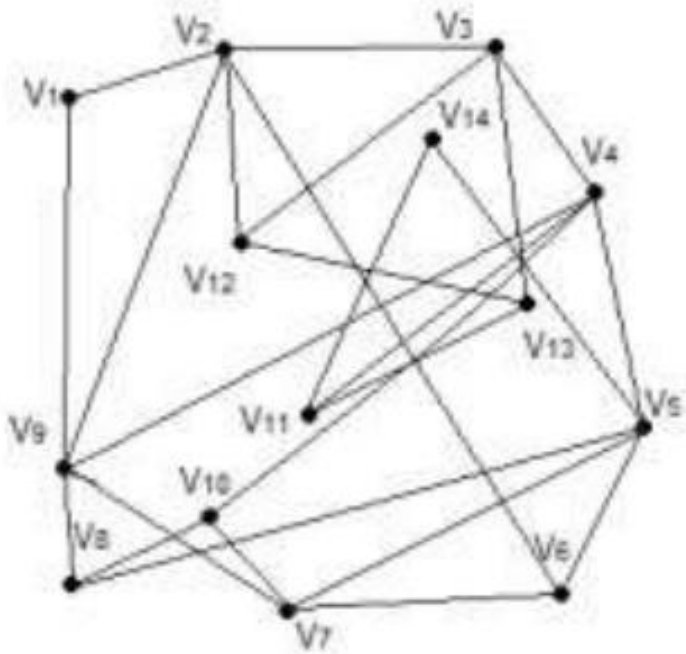
1. За допомогою алгоритму Дейкстри знайти найкоротший шлях у графі поміж парою вершин  $V_0$  і  $V^*$ .

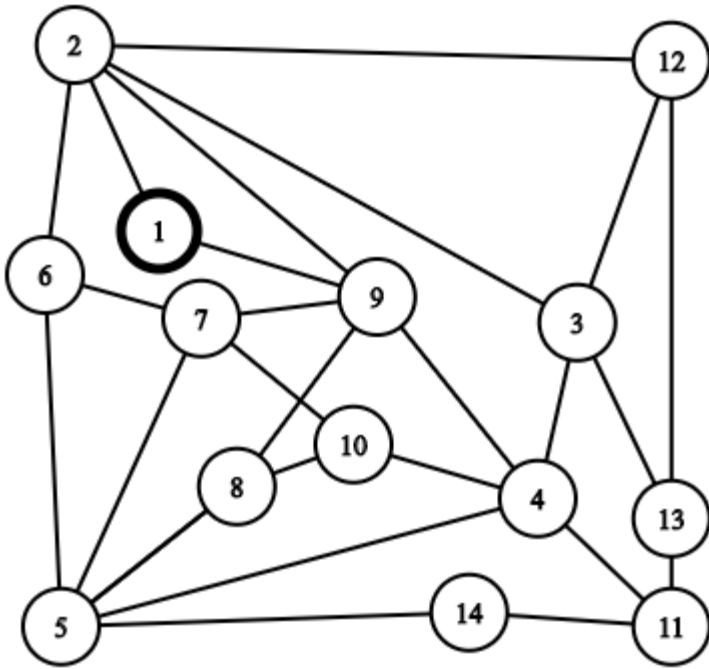




Найкоротша відстань від  $V_0$  до  $V^* = 22$ .

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



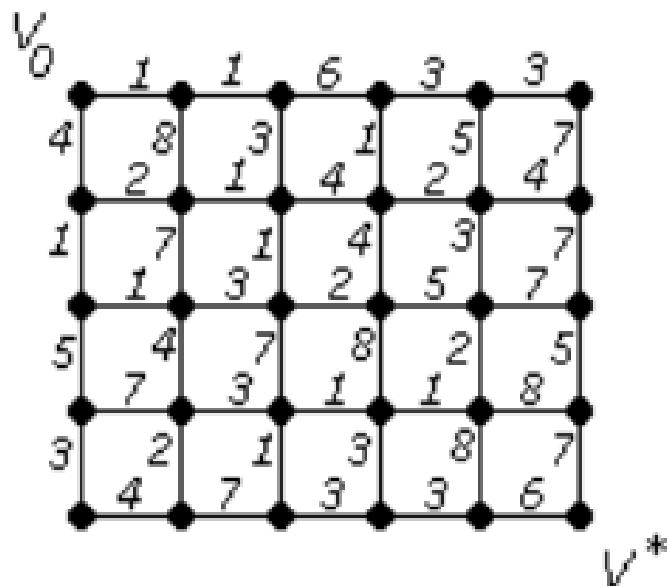


Умовою доведення непланарності графа є наявність п'яťох і більше вершин зі степенем більшим ніж 3, або шістьох і більше вершин зі степенем більшим ніж 2. На даному графі можемо зауважити, як мінімум 5 вершин зі степенем більше 3, зокрема вершини: 3, 4, 5, 7 та 9. Отже граф непланарний та здійснити g -укладку графа на площині неможливо.

Граф неможливо укласти.

## Завдання №2.

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



Код програми:

```
main.cpp
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <iostream>
4
5  using namespace std;
6
7  int min_top(int** arr, int v) {
8      int m = 0;
9      for (int i = 0; i < v; i++)
10         {
11             if (arr[i][1])
12             {
13                 m = i;
14                 break;
15             }
16         }
17         for (int i = 1; i < v; i++)
18         {
19             if (arr[m][0] >= arr[i][0] && arr[i][1] == 1)
20             {
21                 m = i;
22             }
23         }
24         return m;
25     }
26 }
```

```

26 int main()
27 {
28     int inf = 100000;
29     int a, b, c;
30     int v = 0;
31     cout << "Number of Vertices : ";
32     cin >> v;
33     int** graph = new int* [v];
34     for (int j = 0; j < v; j++)
35     {
36         graph[j] = new int[v];
37     }
38     for (int a = 0; a < v; a++)
39     {
40         for (int j = 0; j < v; j++)
41         {
42             graph[a][j] = 0;
43         }
44     }
45     int r = 0;
46     cout << "Enter number of Reber : ";
47     cin >> r;
48     cout << "Enter weight of reber : " << endl;
49     for (size_t i = 0; i < r; i++)
50     {
51         cin >> a;
52         cin >> b;
53         cin >> c;
54         graph[a - 1][b - 1] = graph[b - 1][a - 1] = c;
55     }

```

```

56     int p;
57     int** tops = new int* [v];
58     for (int j = 0; j < v; j++)
59     {
60         tops[j] = new int[2];
61     }
62     int* tops_path = new int[v];
63     cout << "Begin at: ";
64     cin >> p;
65     for (int i = 0; i < v; i++)
66     {
67         if (i == p - 1) {
68             tops[i][0] = 0;
69             tops[i][1] = 1;
70         }
71         else {
72             tops[i][0] = inf;
73             tops[i][1] = 1;
74         }
75     }

```

```

76     tops_path[p - 1] = 0;
77     int m;
78     for (int i = 0; i < v; i++)
79     {
80         m = min_top(tops, v);
81         for (int j = 0; j < v; j++)
82         {
83             if (graph[m][j])
84             {
85                 if (tops[j][0] > tops[m][0] + (graph[m][j]))
86                 {
87                     tops[j][0] = tops[m][0] + (graph[m][j]);
88                     tops_path[j] = m;
89                 }
90             }
91         }
92         tops[m][1] = 0;
93     }
94     //cout << "End at: ";
95     int k;
96     cin >> k;
97     cout << "Shortest route: ";
98     cout << tops[k - 1][0];
99     cout << endl << k << " <-- ";
100     k--;
101     for (int a = 0; tops_path[k] != p - 1; a++)
102     {
103         cout << tops_path[k] + 1 << " <-- ";
104         k = tops_path[k];
105     }

```

```

106 }
107 cout << p << endl;
108 return 0;
109 }
110

```

```
D:\labys\dyskretka\Laba5\Laba5\bin\Debug\Laba5.exe
15 21 7
16 17 5
16 22 8
17 18 7
17 23 2
18 24 5
19 20 7
19 25 3
20 21 3
20 26 2
21 22 1
21 27 1
22 23 1
22 28 3
23 24 8
23 29 8
24 30 7
25 26 4
26 27 7
27 28 3
28 29 3
29 30 6
Begin at: 1
End at: 30
Shortest route: 24
30 <-- 29 <-- 28 <-- 27 <-- 21 <-- 15 <-- 9 <-- 3 <-- 2 <-- 1
Process returned 0 (0x0)   execution time : 15.453 s
Press any key to continue.
```

Висновок: я набув практичних вмінь та навичок з використання алгоритму Дейкстри.