

МІНІСТЕРСТВО ОСВІТИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
«ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра штучного інтелекту

Лабораторна робота №5

З дисципліни

«Дискретна математика»

Виконав:

Студент групи КН-115

Чеханівського Михайла

Викладач:

Мельникова Н.І.

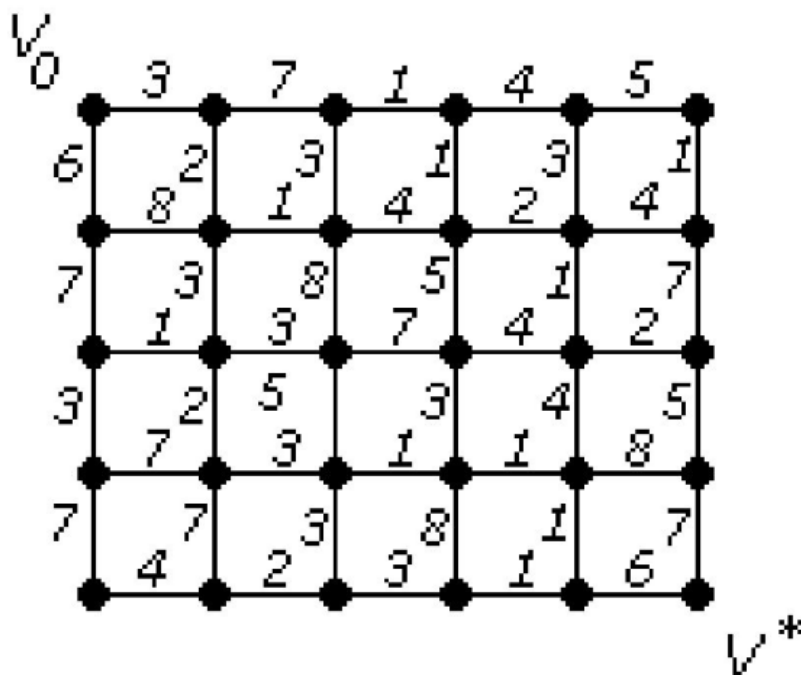
Тема: Знаходження найкоротшого маршруту за алгоритмом Дейкстри. Плоскі планарні графи

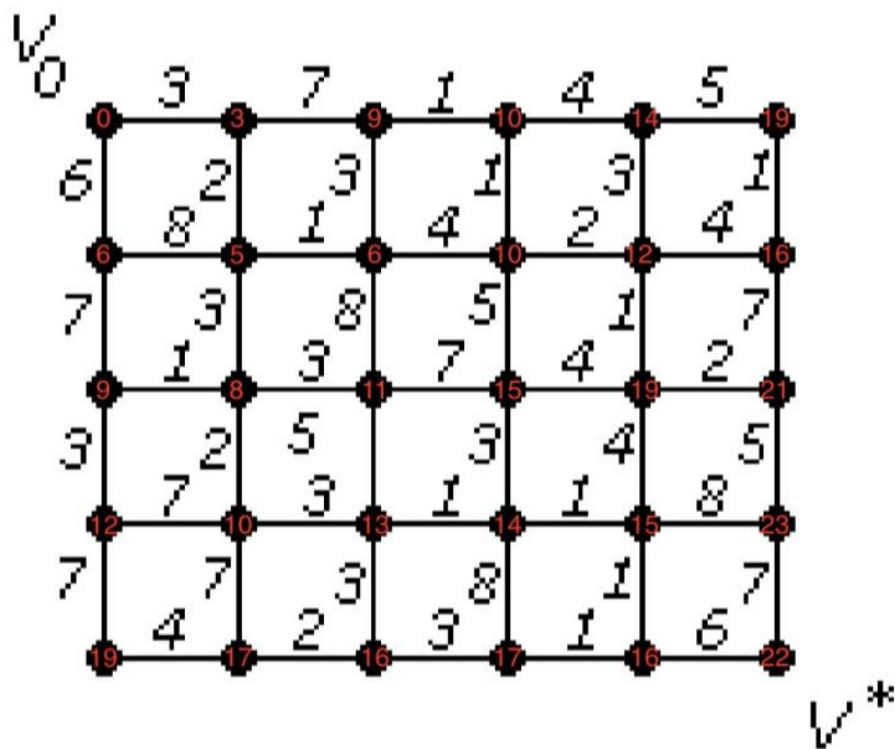
Мета роботи: набуття практичних вмінь та навичок з використання алгоритму Дейкстри.

Варіант 16

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

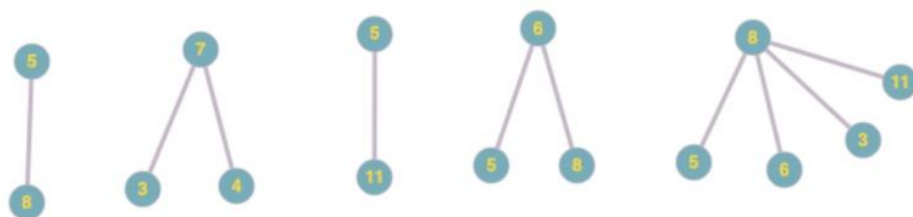
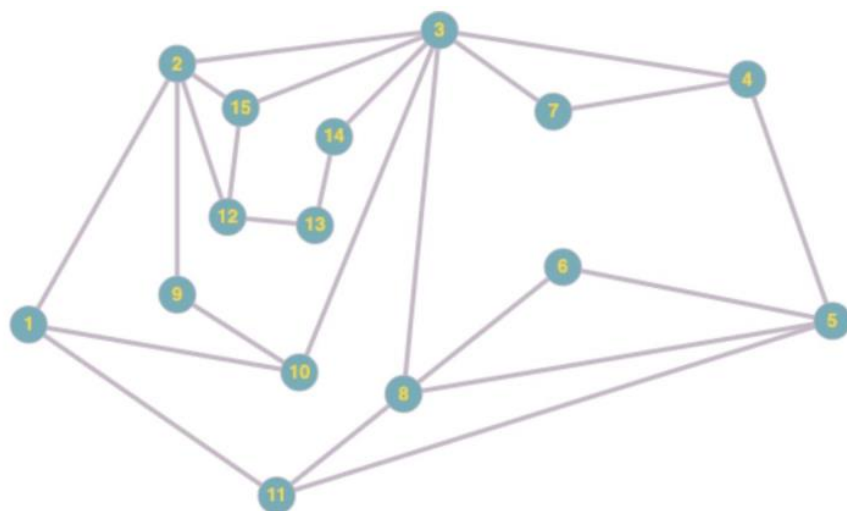
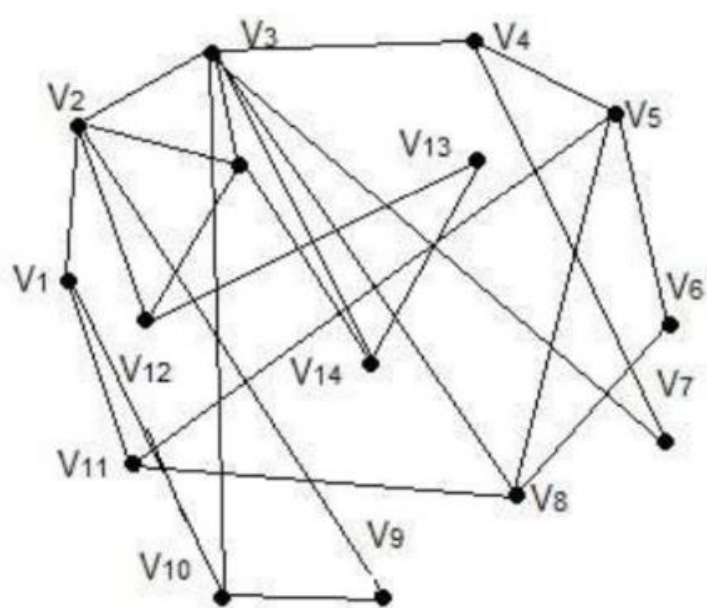
1. За допомогою алгоритму Дейкстра знайти найкоротший шлях у графі поміж парою вершин V_0 і V^* .

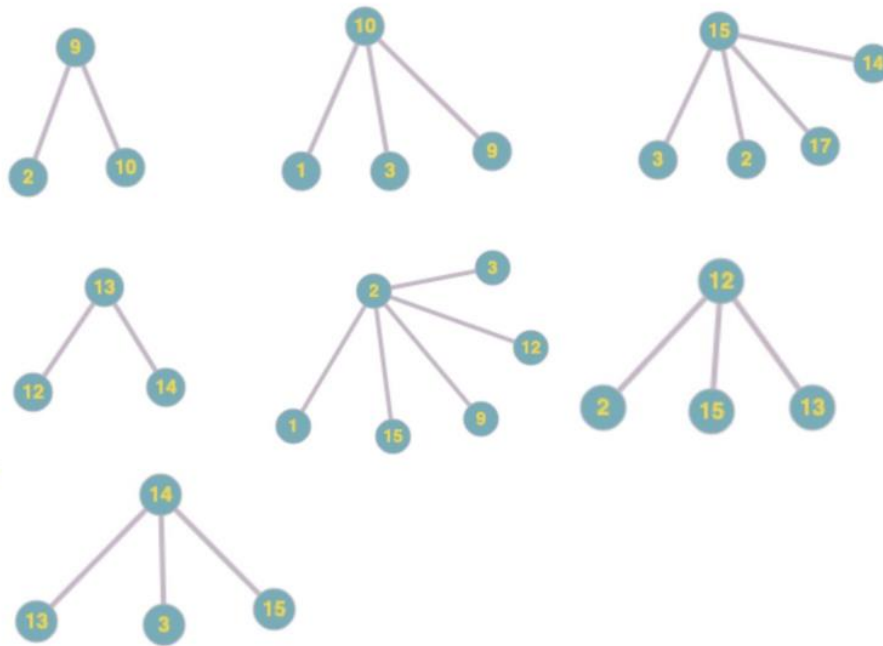




Найкоротший шлях у графі поміж парою вершин V_0 і V^* - 22.

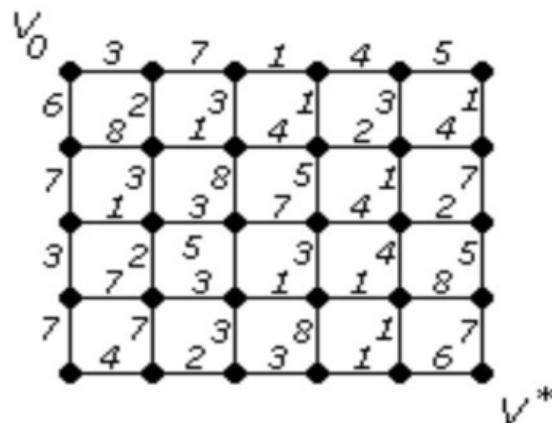
2. За допомогою -алгоритма зробити укладку графа у площині, або довести що вона неможлива.





Завдання №2.

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



```

5   int V, start;
6   int pnode;
7   □int matrix[30][30] = {
8       {0,3,0,0,0,0,6,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
9       {3,0,7,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
10      {0,7,0,1,0,0,0,0,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
11      {0,0,1,0,4,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
12      {0,0,0,4,0,5,0,0,0,0,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
13      {0,0,0,0,5,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
14      {6,0,0,0,0,0,0,8,0,0,0,0,7,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
15      {0,2,0,0,0,0,8,0,1,0,0,0,0,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
16      {0,0,3,0,0,0,0,0,1,0,4,0,0,0,8,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
17      {0,0,0,1,0,0,0,0,4,0,2,0,0,0,0,5,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
18      {0,0,0,0,3,0,0,0,0,2,0,4,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
19      {0,0,0,0,0,1,0,0,0,0,4,0,0,0,0,5,0,7,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
20      {0,0,0,0,0,0,7,0,0,0,0,0,0,1,0,0,0,0,3,0,0,0,0,0,0,0,0,0,0,0,0,0},
21      {0,0,0,0,0,0,0,3,0,0,0,0,1,0,3,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0},
22      {0,0,0,0,0,0,0,0,8,0,0,0,0,0,3,0,7,0,0,0,0,5,0,0,0,0,0,0,0,0,0,0},
23      {0,0,0,0,0,0,0,0,0,5,0,0,0,0,7,0,4,0,0,0,0,3,0,0,0,0,0,0,0,0,0,0},
24      {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},
25      {0,0,0,0,0,0,0,0,0,0,0,7,0,0,0,0,2,0,0,0,0,0,0,5,0,0,0,0,0,0,0,0},
26      {0,0,0,0,0,0,0,0,0,0,0,0,3,0,0,0,0,0,7,0,0,0,0,7,0,0,0,0,0,0,0,0},
27      {0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,7,0,3,0,0,0,7,0,0,0,0,0,0,0},
28      {0,0,0,0,0,0,0,0,0,0,0,0,0,0,5,0,0,0,0,3,0,1,0,0,0,0,3,0,0,0,0,0},
29      {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3,0,0,0,0,1,0,1,0,0,0,0,8,0,0,0,0},
30      {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,4,0,0,0,0,1,0,8,0,0,0,0,1,0,0,0},
31      {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},
32      {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,7,0,0,0,0,0,4,0,0,0,0,0,0},
33      {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,7,0,0,0,0,4,0,2,0,0,0,0},
34      {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3,0,0,0,0,2,0,3,0,0,0},
35      {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,8,0,0,0,0,3,0,1,0,0,0},
36      {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,1,0,6,0,0},
37      {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}
38  };

```

```

39   int sdist[30];
40   bool vis[30] = { 0 };
41   int par[30];
42
43   □int getNear()
44   {
45       int minVal = 999, minNode = 0;
46       □for (int i = 0; i < V; i++)
47       {
48           □if (!vis[i] && sdist[i] < minVal)
49           {
50               minVal = sdist[i];
51               minNode = i;
52           }
53       }
54       return minNode;
55   }
56   □void Deikstra()
57   {
58       □for (int i = 0; i < V; i++)
59       {
60           int near = getNear();
61           vis[near] = true;
62           □for(int j = 0; j < V; j++)
63           {
64               □if (matrix[near][j] != INF && sdist[j] > sdist[near] + matrix[near][j])
65               {
66                   sdist[j] = sdist[near] + matrix[near][j];
67                   par[j] = near;
68               }
69           }
70       }
71   }
72

```

```

73 int main()
74 {
75     V = 30;
76     for (int i = 0; i < V; i++)
77     {
78         for (int j = 0; j < V; j++)
79         {
80             if (matrix[i][j] == 0 && i != j)
81             {
82                 matrix[i][j] = 999;
83             }
84         }
85     }
86     start = 0;
87     for (int i = 0; i < V; i++)
88     {
89         par[i] = i;
90         sdist[i] = INF;
91     }
92     sdist[start] = 0;
93     Deikstra();
94     cout << "Min way: " << sdist[V - 1] << endl;
95     cout << "Way: " << endl << V << " ";
96     pnode = par[V - 1];
97     cout << endl;
98     while (pnode != start)
99     {
100         cout << pnode + 1 << endl;
101         pnode = par[pnode];
102     }
103     cout << "1";
104 }

```