

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

Кафедра систем штучного інтелекту

Лабораторна робота №5

з дисципліни
«Дискретна математика»

Виконала:
студентка групи КН-115
Дзямба Аліна
Викладач:
Мельникова Н. І.

Львів – 2019 р.

Знаходження найкоротшого маршруту за алгоритмом Дейкстри.

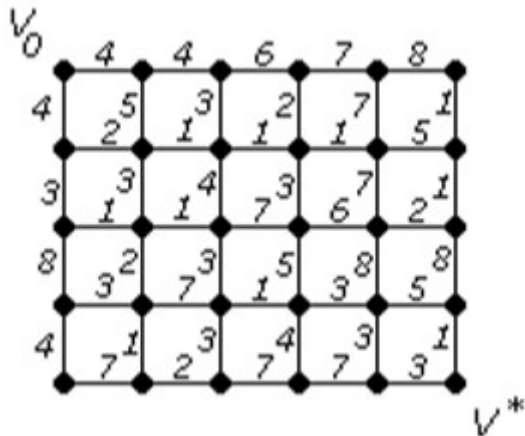
Плоскі планарні граfi

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

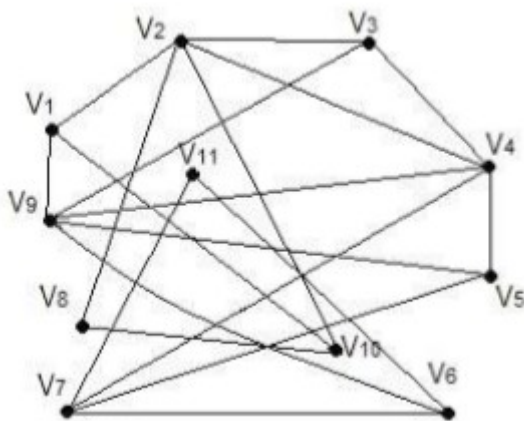
Варіант №7

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^* .

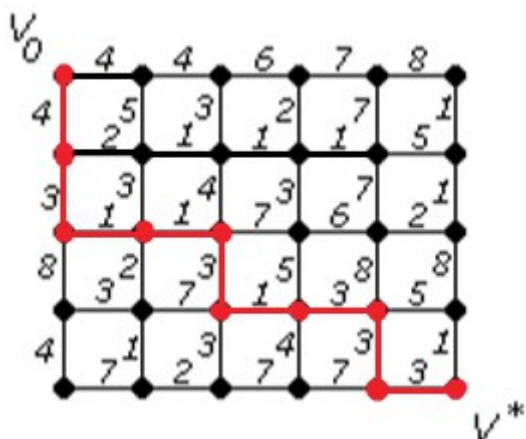


2. За допомогою γ -алгоритма зробити укладку графа у площині, або довести що вона неможлива.



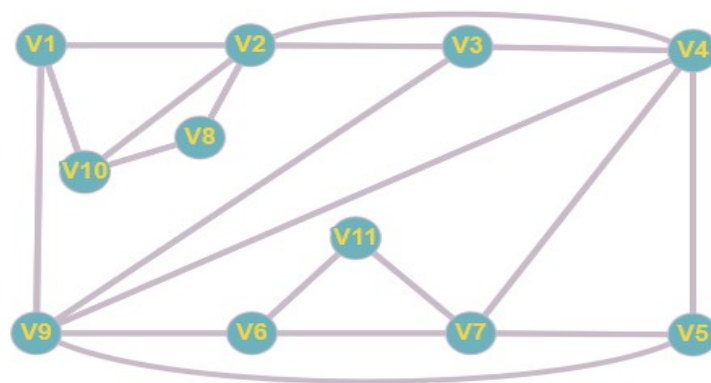
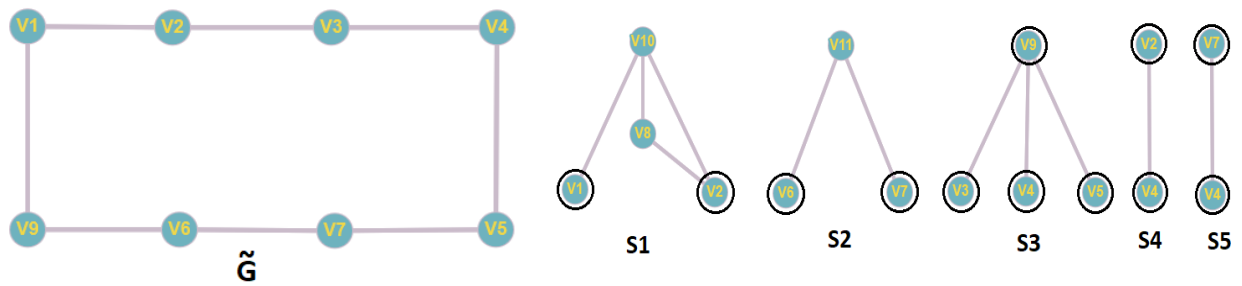
Розв'язання:

1.

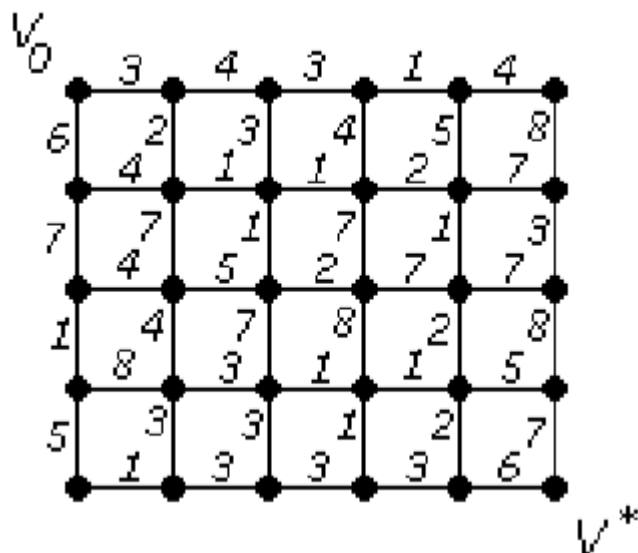


Найкоротша відстань: $4+3+1+1+3+1+3+3+3=22$

2.



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



Програма:

```

1  #include <iostream>
2  #include <stdio.h>
3  using namespace std;
4  #define INFINITY 9999
5  #define max 30
6
7  void algorithm(int G[max][max], int n, int start);
8
9  int main()
10 {
11     int G[max][max] = {
12         {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},
13         {3, 0, 4, 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},
14         {0, 4, 0, 3, 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},
15         {0, 0, 3, 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},
16         {0, 0, 0, 1, 0, 4, 0, 0, 0, 0, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
17         {0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0, 8, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
18         {6, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0, 0, 7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
19         {0, 2, 0, 0, 0, 0, 8, 4, 0, 1, 0, 0, 0, 0, 0, 7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
20         {0, 0, 3, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
21         {0, 0, 0, 4, 0, 0, 0, 0, 1, 0, 2, 0, 0, 0, 0, 7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
22         {0, 0, 0, 0, 5, 0, 0, 0, 0, 2, 0, 7, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
23         {0, 0, 0, 0, 0, 8, 0, 0, 0, 0, 7, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
24         {0, 0, 0, 0, 0, 0, 7, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
25         {0, 0, 0, 0, 0, 0, 0, 7, 0, 0, 0, 0, 4, 0, 5, 0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
26         {0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 5, 0, 2, 0, 0, 0, 0, 7, 0, 0, 0, 0, 0, 0, 0, 0, 0},
27         {0, 0, 0, 0, 0, 0, 0, 0, 0, 7, 0, 0, 0, 0, 2, 0, 7, 0, 0, 0, 8, 0, 0, 0, 0, 0, 0, 0, 0, 0},
28         {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 7, 0, 7, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0},
29         {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 7, 0, 0, 0, 0, 8, 0, 0, 0, 0, 0, 0, 0, 0},
30         {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 8, 0, 0, 0, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
31         {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0, 8, 0, 3, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0},
32         {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 7, 0, 0, 0, 0, 3, 0, 1, 0, 0, 0, 3, 0, 0, 0, 0, 0},
33         {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 8, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0},
34         {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 1, 0, 5, 0, 0, 0, 0, 2, 0, 0, 0},
35         {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 8, 0, 0, 0, 5, 0, 0, 0, 0, 0, 0, 7, 0, 0, 0},
36         {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 5, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0},
37         {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 1, 0, 3, 0, 0, 0, 0, 0},
38         {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 3, 0, 0, 3, 0, 0, 0, 0},
39         {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 3, 0, 3, 0, 0, 0, 0, 0},
40         {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 3, 0, 6, 0, 0, 0, 0, 0},
41         {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}
42     };
43     int n = 30;
44     int u = 0;
45     algorithm(G, n, u);
46     return 0;
47 }
48
49 void algorithm(int G[max][max], int n, int start)
50 {
51     int cost[max][max];
52     int distance[max];
53     int pred[max];
54     int visited[max];
55     int count, min_d, next;
56
57     for (int i = 0; i < n; i++)
58         for (int j = 0; j < n; j++)
59             if (G[i][j] == 0)
60                 cost[i][j] = INFINITY;
61             else
62                 cost[i][j] = G[i][j];
63
64     for (int i = 0; i < n; i++)
65     {
66         distance[i] = cost[start][i];

```

```

67     pred[i] = start;
68     visited[i] = 0;
69 }
70 distance[start] = 0;
71 visited[start] = 1;
72 count = 1;
73
74 while (count < n - 1)
75 {
76     min_d = INFINITY;
77     for (int i = 0; i < n; i++)
78         if (distance[i] < min_d && !visited[i])
79         {
80             min_d = distance[i];
81             next = i;
82         }
83     visited[next] = 1;
84     for (int i = 0; i < n; i++)
85         if (!visited[i])
86             if (min_d + cost[next][i] < distance[i])
87             {
88                 distance[i] = min_d + cost[next][i];
89                 pred[i] = next;
90             }
91     count++;
92 }
93
94 for (int i = 0; i < n; i++)
95     if (i != start)
96     {
97         cout << "\n\tDistance to node " << i << " = " << distance[i];
98         cout << "\nPath = " << i;
99         int j = i;

```

```

100         do
101         {
102             j = pred[j];
103             cout << " <- " << j;
104         } while (j != start);
105     }
106 }

```

Результат програми:

```
Distance to node 1 = 3
Path = 1 <- 0
Distance to node 2 = 7
Path = 2 <- 1 <- 0
Distance to node 3 = 10
Path = 3 <- 2 <- 1 <- 0
Distance to node 4 = 11
Path = 4 <- 3 <- 2 <- 1 <- 0
Distance to node 5 = 15
Path = 5 <- 4 <- 3 <- 2 <- 1 <- 0
Distance to node 6 = 6
Path = 6 <- 0
Distance to node 7 = 5
Path = 7 <- 1 <- 0
Distance to node 8 = 6
Path = 8 <- 7 <- 1 <- 0
Distance to node 9 = 7
Path = 9 <- 8 <- 7 <- 1 <- 0
Distance to node 10 = 9
Path = 10 <- 9 <- 8 <- 7 <- 1 <- 0
Distance to node 11 = 16
Path = 11 <- 10 <- 9 <- 8 <- 7 <- 1 <- 0
Distance to node 12 = 13
Path = 12 <- 6 <- 0
Distance to node 13 = 12
Path = 13 <- 7 <- 1 <- 0
Distance to node 14 = 7
Path = 14 <- 8 <- 7 <- 1 <- 0
Distance to node 15 = 9
Path = 15 <- 14 <- 8 <- 7 <- 1 <- 0
Distance to node 16 = 10
Path = 16 <- 10 <- 9 <- 8 <- 7 <- 1 <- 0
Distance to node 17 = 17
Path = 17 <- 16 <- 10 <- 9 <- 8 <- 7 <- 1 <- 0
Distance to node 18 = 14
Path = 18 <- 12 <- 6 <- 0
Distance to node 19 = 16
Path = 19 <- 13 <- 7 <- 1 <- 0
Distance to node 20 = 14
Path = 20 <- 14 <- 8 <- 7 <- 1 <- 0
Distance to node 21 = 13
Path = 21 <- 22 <- 16 <- 10 <- 9 <- 8 <- 7 <- 1 <- 0
```

```
Path = 22 <- 16 <- 10 <- 9 <- 8 <- 7 <- 1 <- 0
Distance to node 23 = 17
Path = 23 <- 22 <- 16 <- 10 <- 9 <- 8 <- 7 <- 1 <- 0
Distance to node 24 = 19
Path = 24 <- 18 <- 12 <- 6 <- 0
Distance to node 25 = 19
Path = 25 <- 19 <- 13 <- 7 <- 1 <- 0
Distance to node 26 = 17
Path = 26 <- 20 <- 14 <- 8 <- 7 <- 1 <- 0
Distance to node 27 = 14
Path = 27 <- 21 <- 22 <- 16 <- 10 <- 9 <- 8 <- 7 <- 1 <- 0
Distance to node 28 = 14
Path = 28 <- 22 <- 16 <- 10 <- 9 <- 8 <- 7 <- 1 <- 0
Distance to node 29 = 20
Path = 29 <- 28 <- 22 <- 16 <- 10 <- 9 <- 8 <- 7 <- 1 <- 0
```

Висновок: під час виконання лабораторної роботи я набула практичних вмінь та навичок з використання алгоритму Дейкстри.