

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту

Лабораторна робота №5

з дисципліни

«Дискретна математика»

Виконала:

студентка групи КН-112
Казьоннікова Ніколетта

Викладач: Мельникова Н.І.

Львів – 2019 р.

Тема: Знаходження найкоротшого маршруту за алгоритмом

Дейкстри. Плоскі планарні графи

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

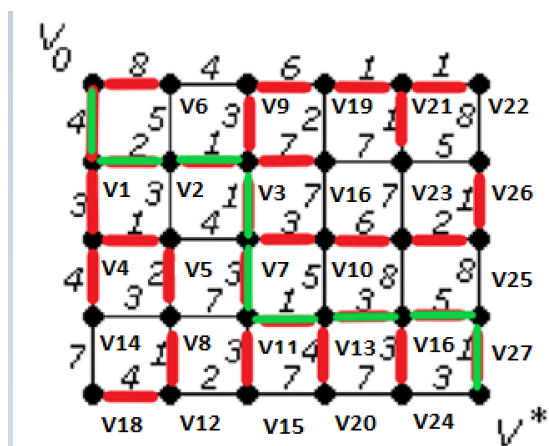
Варіант 8

ІНДИВІДУАЛЬНІ ЗАВДАННЯ

Завдання №1

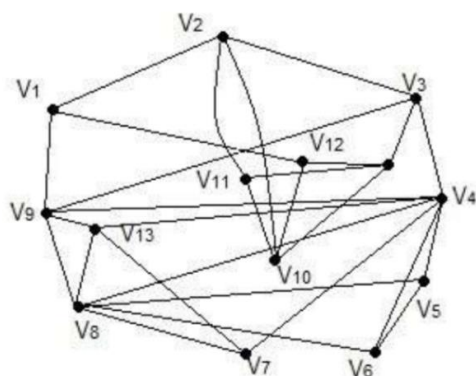
Розв'язати на графах наступні 2 задачі:

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

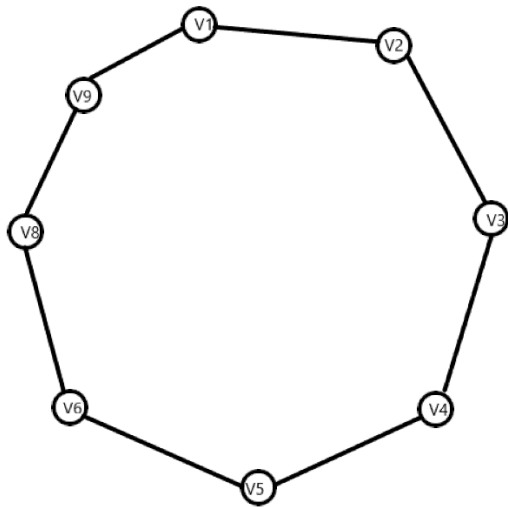


$V_1=4; V_2=6; V_3=7; V_4=7; V_5=8; V_6=8; V_7=8; V_8=10; V_9=10; V_{10}=11; V_{11}=11; V_{12}=12;$
 $V_{13}=12; V_{14}=12; V_{15}=14; V_{16}=14; V_{17}=15; V_{18}=16; V_{19}=16; V_{20}=16; V_{21}=17;$
 $V_{22}=18; V_{23}=18; V_{24}=18; V_{25}=19; V_{26}=20; V_{27}=20; V^*=2$

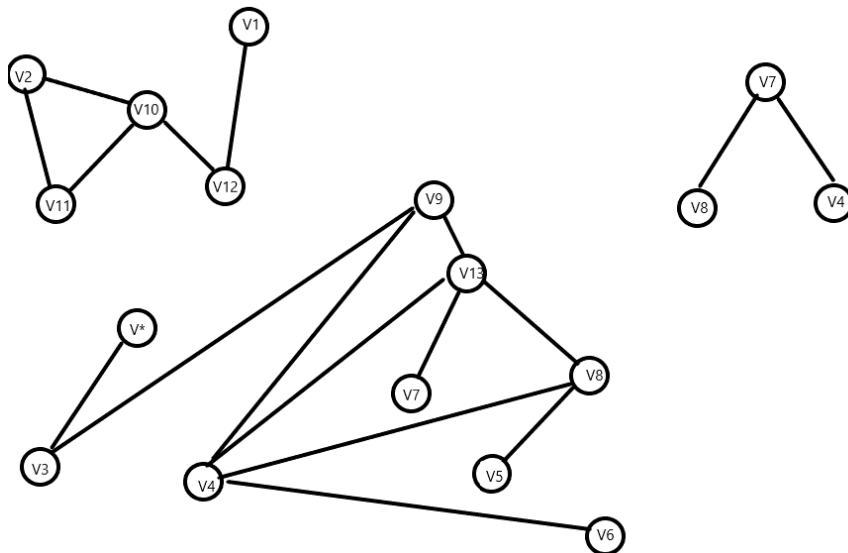
2. За допомогою у-алгоритма зробити укладку графа у площині, або довести що вона неможлива.



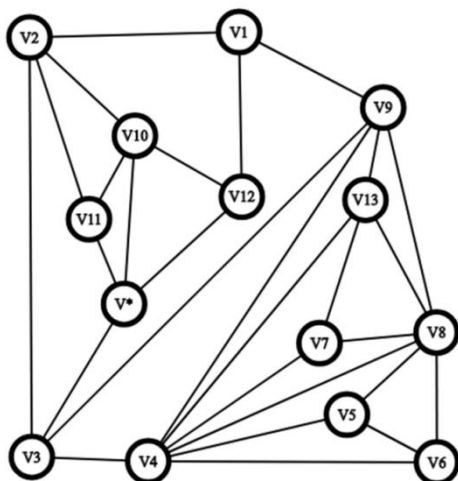
Обираємо найбільший цикл графа



І залишки



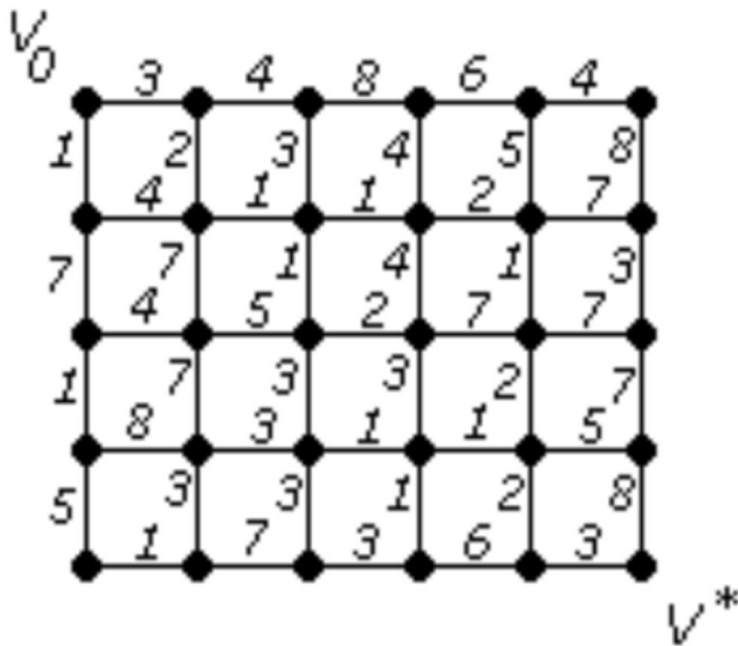
Результат:



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:

```

1  #define _CRT_SECURE_NO_WARNINGS
2  #include <iostream>
3  #include <fstream>
4  #include <cstring>
5  #include <cstdio>
6  #include <sstream>
7  using namespace std;
8
9  const int number = 30;
10 const int mwei = 1000;
11
12 struct vershuna
13 {
14     int weight = mwei;
15     int from = 0;
16     bool fixed = false;
17 };
18
19 int main()
20 {
21     int matrix[number][number];
22     string rebro;
23     char rebroch[100];
24     char* rch;
25     vershuna ver[number];
26     ver[0].weight = 0;
27     ver[0].fixed = true;
28     int curver = 0;
29     for (int i = 0; i < number; i++)
30     {
31         for (int j = 0; j < number; j++)
32         {

```

```

33         matrix[i][j] = 0; //zanululu matrix
34     }
35 }
36 int i = 0; int j = 0; int row = 0;
37 fstream file;
38 file.open("MATRIX.txt", ios_base::in); // zapus iz file
39 while (getline(file, rebro))
40 {
41     strcpy(rebroch, rebro.c_str());
42     char* rch = strtok(rebroch, " ");
43     while (rch != NULL)
44     {
45         if (row % 2 == 0)
46         {
47             matrix[i][i + 1] = (int)*rch - 48;
48             matrix[i + 1][i] = (int)*rch - 48;
49             i++;
50         }
51         else
52         {
53             matrix[j][j + 6] = (int)*rch - 48;
54             matrix[j + 6][j] = (int)*rch - 48;
55             j++;
56         }
57         rch = strtok(NULL, " ");
58     }
59     row++;
60     if (row % 2 == 0) i++;
61 }
62 for (int i = 0; i < number; i++)
63 {
64     for (int j = 0; j < number; j++)
65     {
66         cout << matrix[i][j] << " "; //druk matrix
67     }
68     cout << endl;
69 }
70 //alhoritm d
71
72 while (ver[number - 1].fixed != true)
73 {
74     int conected[number - 1];
75     int index = 0; //find sumizhni vershunu dlya currver
76     for (int i = 0; i < number; i++)
77     {
78         if (matrix[currver][i] != 0)
79         {
80             conected[index] = i;
81             index++;
82         }
83     }
84
85     for (int i = 0; i < index; i++) //dlya sumizhnuch ver find vaha
86     {
87         if ((ver[currver].weight + matrix[currver][conected[i]]) < ver[conected[i]].weight)
88         {
89             ver[conected[i]].weight = ver[currver].weight + matrix[currver][conected[i]];
90             ver[conected[i]].from = currver;
91         }
92     }
93
94     int min = mwei; //znach ver iz min vaha i fix ii
95     int ind = 0;

```

```

96     for (int i = 0; i < number; i++)
97     {
98         if ((ver[i].weight < min) && (ver[i].fixed != true))
99         {
100             min = ver[i].weight;
101             ind = i;
102         }
103     }
104     ver[ind].fixed = true;
105     currver = ind; //prodovzh itu po ver z min vahoyu
106 }
107 //druk
108 int track[number];
109 int trindex = 0;
110 currver = number - 1;
111 track[trindex] = currver;
112 trindex++;
113 while (currver > 0)
114 {
115     track[trindex] = ver[currver].from;
116     trindex++;
117     currver = ver[currver].from;
118 }
119 cout << "\nMin track(" << ver[number - 1].weight << "): ";
120 for (int i = trindex - 1; i >= 0; i--)
121 {
122     cout << track[i] + 1;
123     if (i != 0) cout << "-";
124     else cout << endl;
125 }
126 file.close();
127 }

```

Результати програми:

[illegible]

Висновки: в ході лабораторної ми набули практичні вміння та навички з використання алгоритму Дейкстри.