

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

Кафедра систем штучного інтелекту

Лабораторна робота №_5

з дисципліни

«Дискретна математика»

Виконала:

Студентка

групи КН-112

Гудз Юлія

Викладач:

Мельникова Н.І.

Львів – 2019 р.

Лабораторна робота № 5.

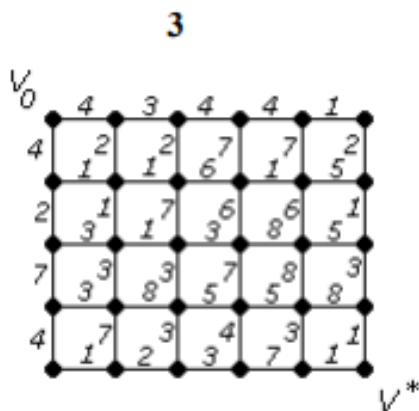
Тема: Знаходження найкоротшого маршруту за алгоритмом Дейкстри. Плоскі планарні граfi

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

Варіант № 3 :

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

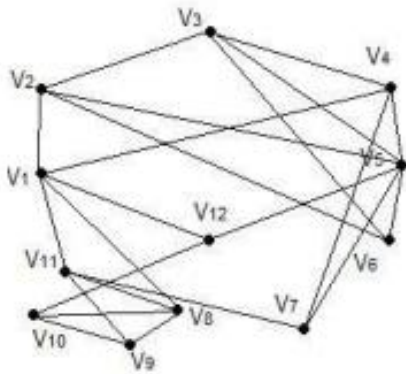
1. За допомогою алгоритму Дейкстра знайти найкоротший шлях у графі поміж парою вершин V_0 і V^* .



$V_1=4$; $V_2=4$; $V_3=5$; $V_4=6$; $V_5=6$; $V_6=6$; $V_7=7$; $V_8=7$; $V_9=9$; $V_{10}=10$; $V_{11}=10$; $V_{12}=11$; $V_{13}=12$; $V_{14}=12$; $V_{15}=13$; $V_{16}=13$; $V_{17}=15$; $V_{18}=15$; $V_{19}=15$; $V_{20}=16$; $V_{21}=16$; $V_{22}=16$; $V_{23}=18$; $V_{24}=18$; $V_{25}=19$; $V_{26}=20$; $V_{27}=22$; $V_{28}=23$; $V_{29}=23$

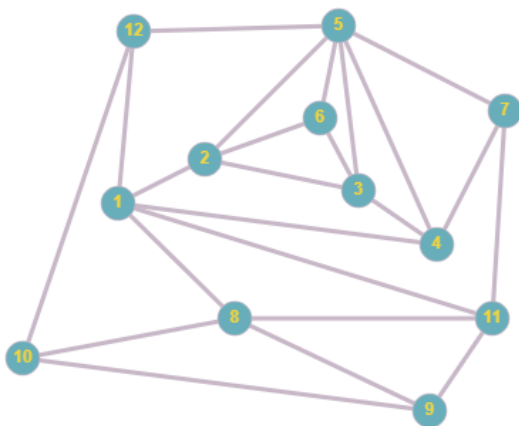
2. За допомогою у-алгоритма зробити укладку графа у площині, або довести що вона неможлива.

3

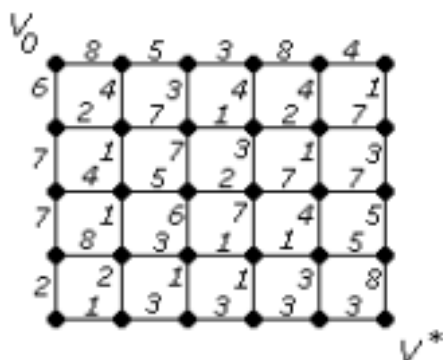


Виділимо граф $V_1, V_2, V_3, V_4, V_7, V_{11}$

Всі інші вершини виносимо назовні



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



```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     setlocale(LC_ALL, "Ukrainian");
6     int v;
7     cout << "Кількість вершин графа : ";
8     cin >> v;
9     int graph[v][v];
10    for (int i = 0; i < v; i++) {
11        for (int j = 0; j < v; j++) {
12            graph[i][j] = 0;
13        }
14    }
15    int a, b, c;
16    cout << "Заповнити таблицю (мож заповнити вершину 0)" << endl;
17    cout << "Вершина 1 | Вершина 2 | Вартість : " << endl;
18    while (1) {
19        cin >> a;
20        if (a == 0) {
21            break;
22        }
23        cin >> b >> c;
24        graph[a - 1][b - 1] = graph[b - 1][a - 1] = c;
25    }
26    int verts[v][2];
27    int steps[v];
28
29    int first = 1;
30    for (int i = 0; i < v; i++) {
31        if (i == first - 1) {
32            verts[i][0] = 0;
33            verts[i][1] = 1;
34        }
35        else {
36            verts[i][0] = 999;
37            verts[i][1] = 1;
38        }
39    }
40    steps[first - 1] = 0;
41
42    int m;
43
44    for (int i = 0; i < v; i++) {
45        for (int j = 0; j < v; j++) {
46            if (verts[i][1]) {
47                m = i;
48                break;
49            }
50        }
51        for (int i = 1; i < v; i++) {
52            if (verts[m][0] >= verts[i][0] && verts[i][1] == 1) {
53                m = i;
54            }
55        }
56        for (int j = 0; j < v; j++) {
57            if (graph[m][j]) {
58                if (verts[j][0] > verts[m][0] + (graph[m][j])) {
59                    verts[j][0] = verts[m][0] + (graph[m][j]);
60                    steps[j] = m;
61                }
62            }
63        }
64        verts[m][1] = 0;
65    }
66
67    int last = 30;
68    cout << "Мінімальний шлях: ";
69    cout << verts[last - 1][0];
70    cout << endl << last << " ";
71    last--;
72    while (steps[last] != first - 1) {
73        cout << steps[last] + 1 << " ";
74        last = steps[last];
75    }
76    cout << first << endl;
77    return 0;
78 }
```

Результат програми:

```
C:\Users\user\Labs\laba.dm.5\bin\Debug\laba.dm.exe
Кількість вершин графа : 30
Заповніть таблицю(щоб завершити введіть 0)
|вершина 1|вершина 2|вага| :
1 2 8
2 3 5
3 4 3
4 5 8
5 6 4
7 8 2
8 9 7
9 10 1
10 11 2
11 12 7
13 14 4
14 15 5
15 16 2
16 17 7
17 18 7
19 20 8
20 21 3
21 22 1
22 23 1
23 24 5
25 26 1
26 27 3
27 28 3
28 29 3
29 30 3
1 7 6
2 8 4

3 9 3
4 10 4
5 11 4
6 12 1
7 13 7
8 14 1
9 15 7
10 16 3
11 17 1
12 18 3
13 19 7
14 20 1
15 21 6
16 22 7
17 23 4
18 24 5
19 25 2
20 26 2
21 27 1
22 28 1
23 29 3
24 30 8
0
Мінімальний шлях: 21
30; 29; 28; 22; 21; 20; 14; 8; 7; 1

Process returned 0 (0x0)   execution time : 16.674 s
Press any key to continue.
```