

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНОМУ
УНІВЕРСИТЕТІ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**

Кафедра систем штучного інтелекту

Лабораторна робота №5

з дисципліни
«Дискретна математика»

Виконала:

Студентка групи КН-114

Огорілко Вікторія

Викладач:

Мельникова Н.І.

Львів – 2019р.

Лабораторна робота №5

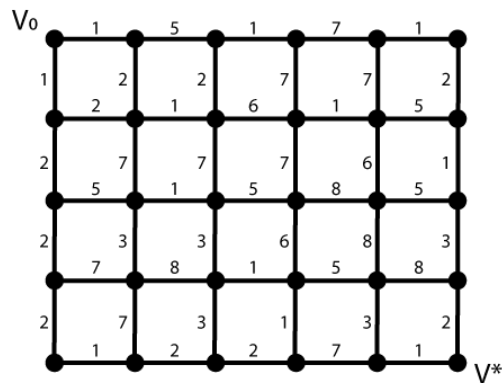
Тема: Знаходження найкоротшого маршруту за алгоритмом Дейкстри. Плоскі планарні графи

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

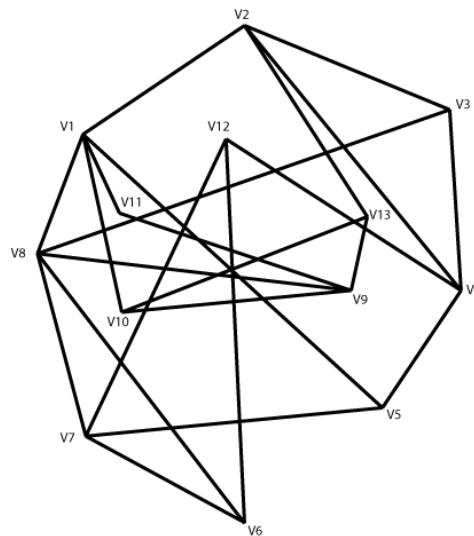
Завдання з Додатка.1

Варіант №1

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

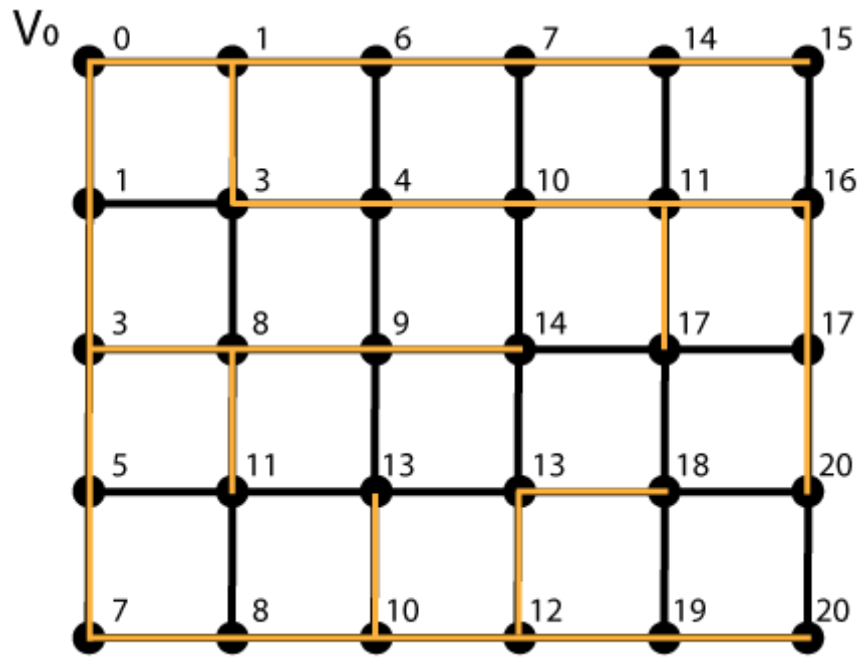


2. За допомогою γ -алгоритма зробити укладку графа у площині, або довести що вона неможлива.



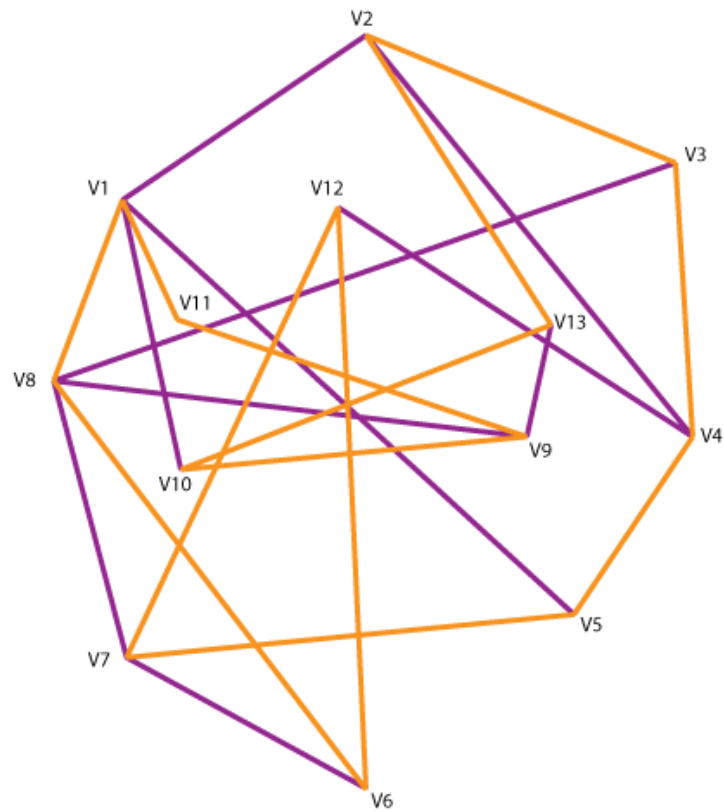
Розв'язування:

1.

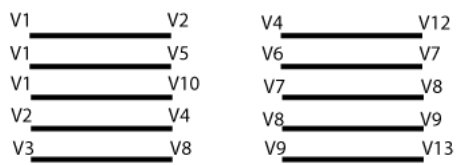
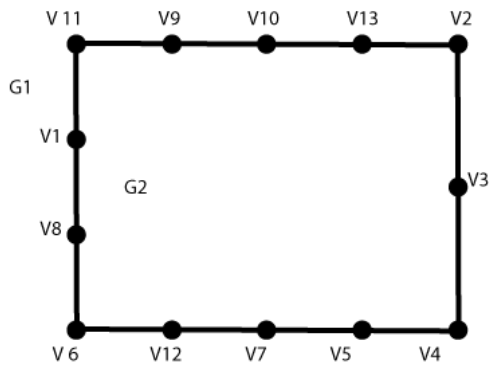


Вага найкоротшого маршруту : 20.

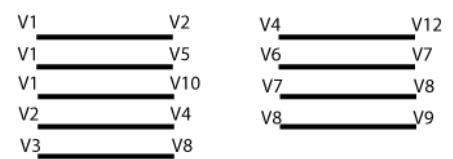
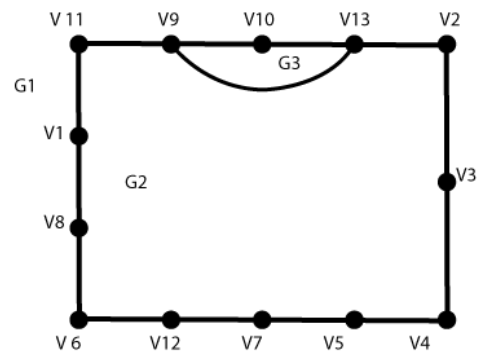
2. Виділимо цикл для початку укладання графа.



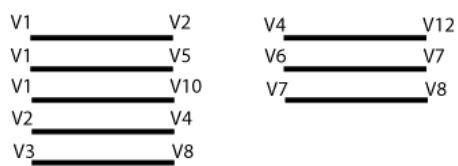
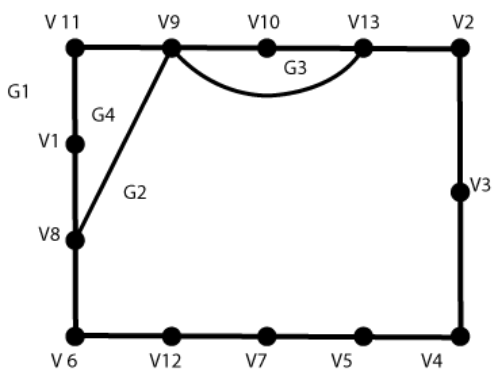
1.



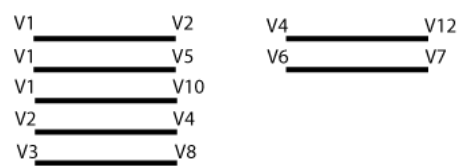
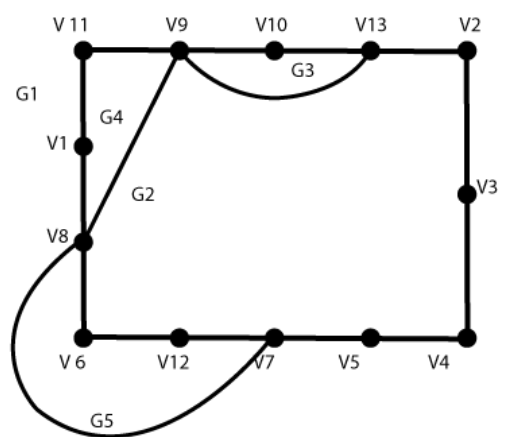
2.



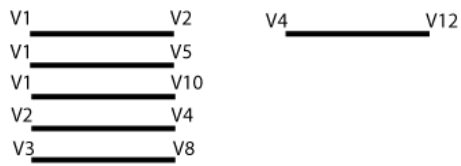
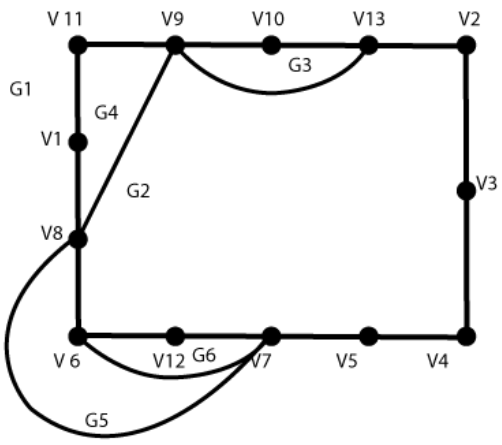
2.



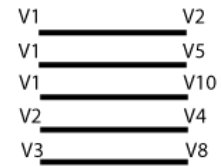
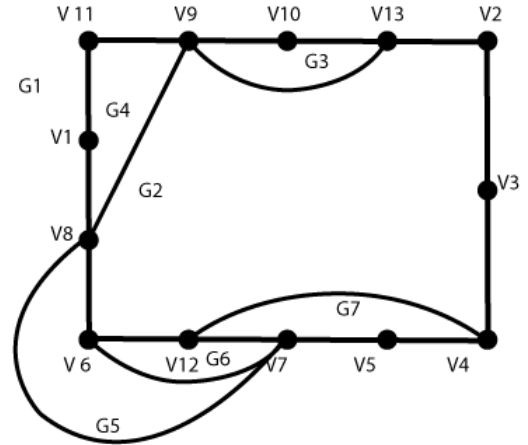
4.



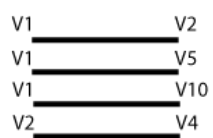
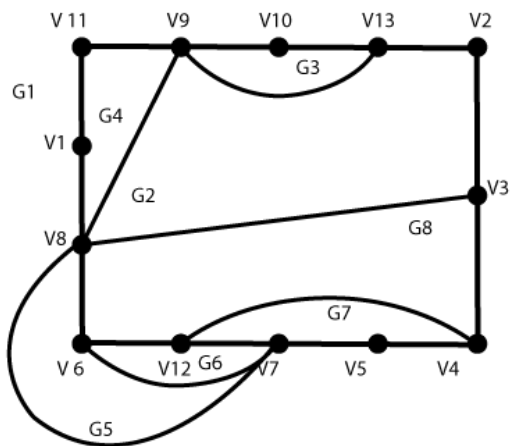
5.



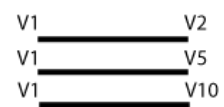
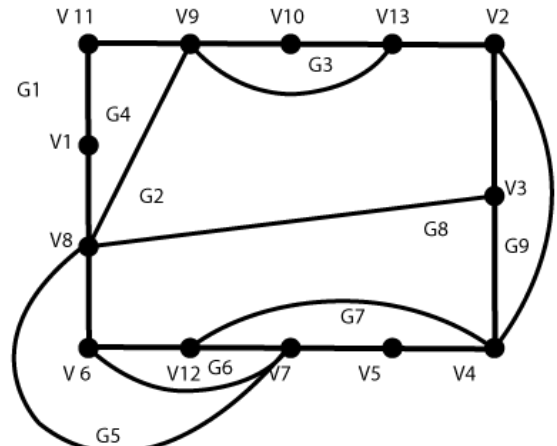
6.



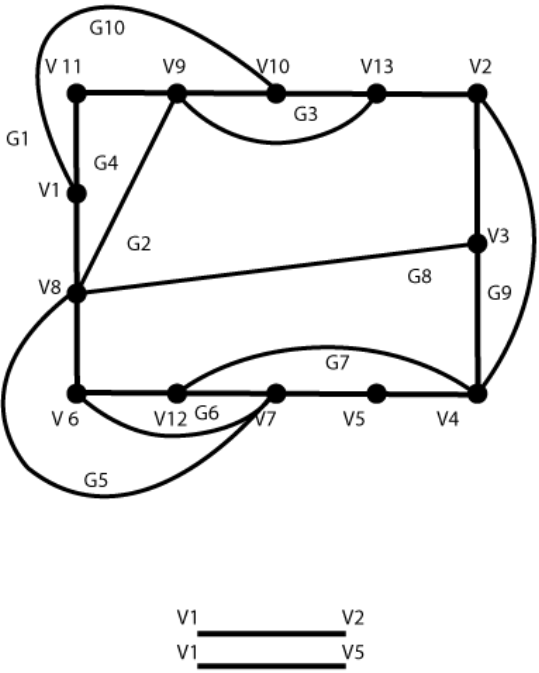
7.



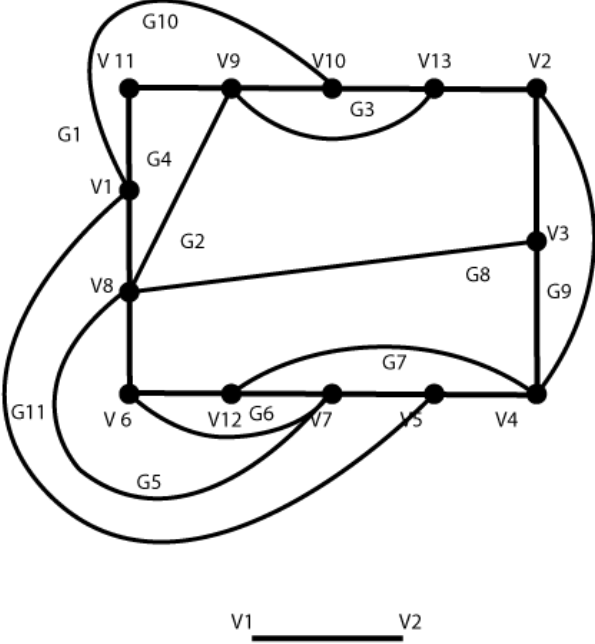
8.



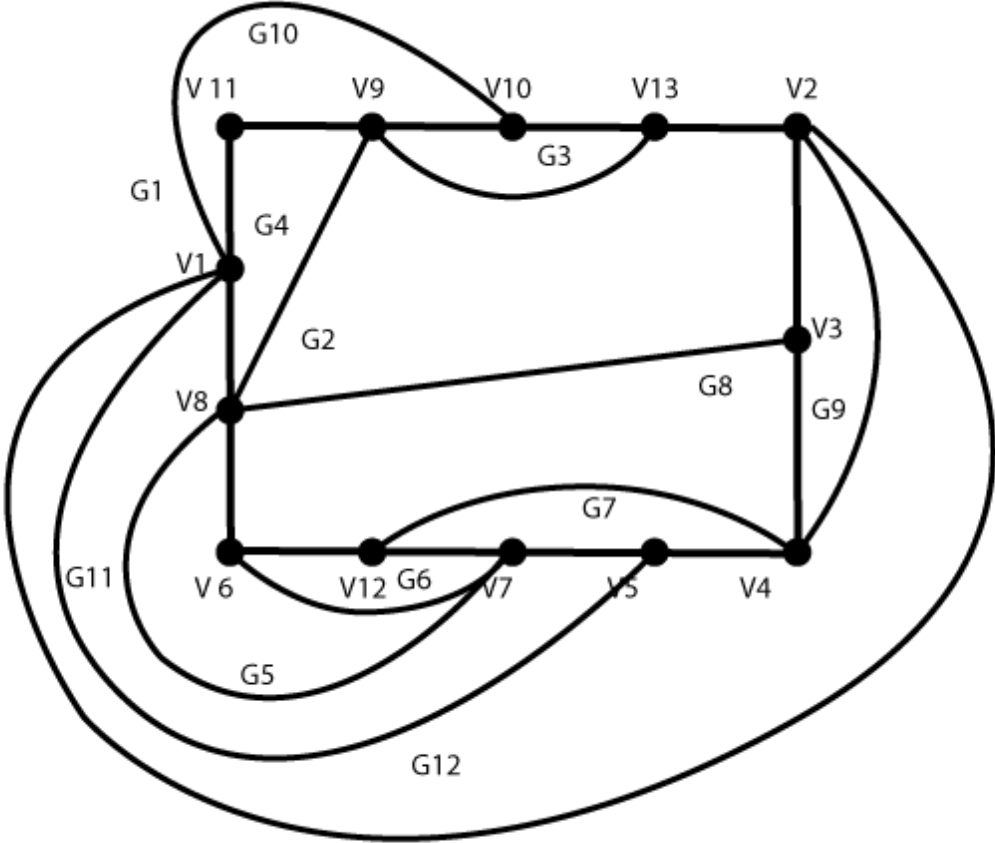
9.



10.

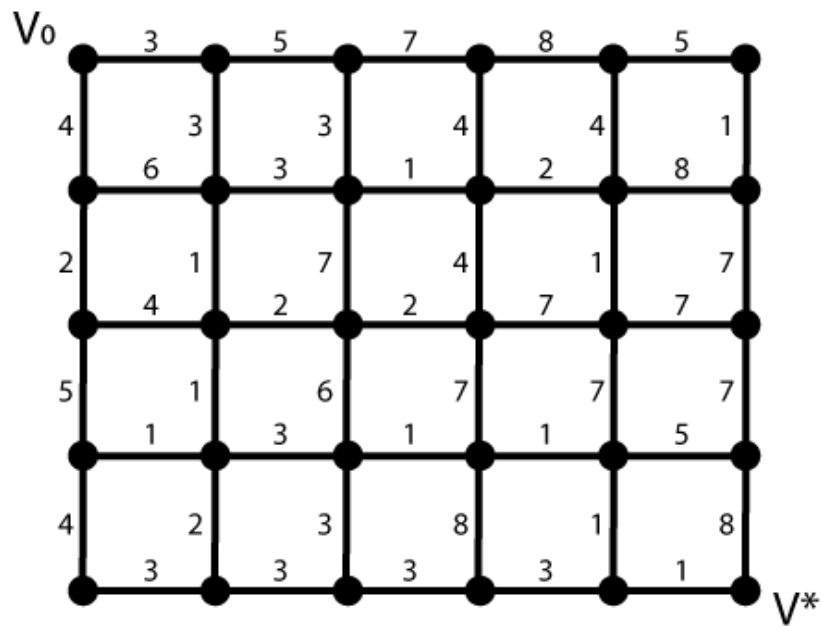


11.



Завдання з Додатка.2

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



Код програми:

```
1  #include <iostream>
2  #include <cstring>
3  using namespace std;
4
5  string hid(int a) {
6      string one = "", two = "V(";
7      int b;
8      do {
9          b = a % 10;
10         switch (b) {
11             case 0:
12                 one = "0" + one; break;
13             case 1:
14                 one = "1" + one; break;
15             case 2:
16                 one = "2" + one; break;
17             case 3:
18                 one = "3" + one; break;
19             case 4:
20                 one = "4" + one; break;
21             case 5:
22                 one = "5" + one; break;
23             case 6:
24                 one = "6" + one; break;
25             case 7:
26                 one = "7" + one; break;
27             case 8:
28                 one = "8" + one; break;
29             case 9:
30                 one = "9" + one; break;
31             default: cout << "error";
32         }
33         a /= 10;
34     } while (a != 0);
35     two += one + ")";
36     return two;
37 }
38
39 int main()
40 {
41     string* vuvid;
42     int n;
43     cout << "\nenter number of all vertices : ";
44     cin >> n;
45     int n0;
46     cout << "\nenter number of vertices in the first line : ";
47     cin >> n0;
48     cout << "\nnnow\nenter data in order :\nA(1)  A(2)  A(1)A(2)  (weight of the edge between them)\n";
49     long int** arr, * arr1, * arr2;
50     arr = new long int* [n];
51     arr1 = new long int[n];
52     arr2 = new long int[n];
53     vuvid = new string[n];
54     for (int i = 0; i < n; i++) {
55         arr[i] = new long int[n];
56         arr2[i] = 0;
57         arr1[i] = 0;
58         for (int j = 0; j < n; j++) {
59             arr[i][j] = 0;
```



```

60     }
61 }
62 int y=0, o=0;
63 int a1, a2, w;
64 for (int i = 0; i < n; i++) {
65     for (int j = 0; j < n; j++) {
66         a1 = i + 1;
67         a2 = j + 1;
68         if (((a2 - a1 == 1) && (a1 % n0 != 0)) || (a2 - a1 == n0)) {
69             cout << hid(a1) << "\t" << hid(a2) << "\t:\t";
70             cin >> w;
71             arr[a1 - 1][a2 - 1] = w;
72             arr[a2 - 1][a1 - 1] = w;
73         }
74     }
75 }
76
77 arr1[0] = 1;
78 vuvid[0] = hid(1);
79 int k = 0, z=0,f=0;
80 long int* ww;
81 ww = new long int[n];
82 ww[0] = 0;
83 for (int i = 0; k != n-1; i++) {
84     y = ww[z];
85     z++;
86     if(y!=n-1){
87         for (int j = 0; j < n; j++) {
88             if (arr[j][y] != 0) {
89                 f++;
90                 ww[f] = j;
91                 if (arr2[j] == 0) {
92                     arr2[j] = (arr[y][j]+arr2[y]);
93                     vuvid[j] = vuvid[y] + " -> " + hid(j + 1);
94                 }
95                 else {
96                     if (arr2[j] > (arr[y][j] + arr2[y])) {
97                         arr2[j] = (arr[y][j] + arr2[y]);
98                         vuvid[j] = vuvid[y] + " -> " + hid(j + 1);
99                     }
100                 }
101                 arr[y][j] = 0;
102                 arr[j][y] = 0;
103             }
104         }
105         arr1[y] = 2;
106         k = 0;
107         for (int j = 0; j < n; j++) {
108             if (arr1[j] == 2) { k++; }
109         }
110     }
111 }
112 cout <<endl<< vuvid[n - 1] << " = " << arr2[n - 1]<<"\n\n\n\n";
113 return 0;
114 }

```

Вивід програми:

```
enter number of all vertices : 30
enter number of vertices in the first line : 6
now
enter data in order :
A(1)  A(2)  A(1)A(2)  (weight of the edge between them)
V(1)  V(2)  :      3
V(1)  V(7)  :      4
V(2)  V(3)  :      5
V(2)  V(8)  :      3
V(3)  V(4)  :      7
V(3)  V(9)  :      3
V(4)  V(5)  :      8
V(4)  V(10) :      4
V(5)  V(6)  :      5
V(5)  V(11) :      4
V(6)  V(12) :      1
V(7)  V(8)  :      6
V(7)  V(13) :      2
V(8)  V(9)  :      3
V(8)  V(14) :      1
V(9)  V(10) :      1
V(9)  V(15) :      7
V(10) V(11) :      2
V(10) V(16) :      4
V(11) V(12) :      8
V(11) V(17) :      1
V(12) V(18) :      7
V(13) V(14) :      4
V(13) V(19) :      5
V(14) V(15) :      2
V(14) V(20) :      1
V(15) V(16) :      2
V(15) V(21) :      6
V(16) V(17) :      7
V(16) V(22) :      7
V(17) V(18) :      7
V(17) V(23) :      7
V(18) V(24) :      7
V(19) V(20) :      1
V(19) V(25) :      4
V(20) V(21) :      3
V(20) V(26) :      2
V(21) V(22) :      1
V(21) V(27) :      3
V(22) V(23) :      1
V(22) V(28) :      8
V(23) V(24) :      5
V(23) V(29) :      1
V(24) V(30) :      8
V(25) V(26) :      3
V(26) V(27) :      3
V(27) V(28) :      3
V(28) V(29) :      3
V(29) V(30) :      1

V(1) -> V(2) -> V(8) -> V(14) -> V(20) -> V(21) -> V(22) -> V(23) -> V(29) -> V(30) = 15
```

Висновок: Я набула практичні вміння та навички з використання алгоритму Дейкстри та виконання гамма укладки графа на площині.