МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Інститут комп'ютерних наук та інформаційних технологій

Кафедра систем штучного інтелекту

Лабораторна робота №5

З дисципліни

«Дискретна математика»

Виконав:

студент групи КН-113

Рябчук Андрій

Викладач:

Мельникова Н.І.

Львів-2019

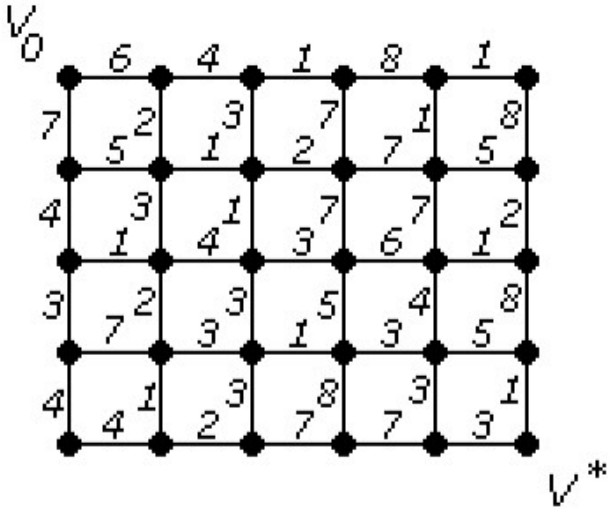
Тема: Знаходження найкоротшого маршруту за алгоритмом Дейкстри. Плоскі планарні графи

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

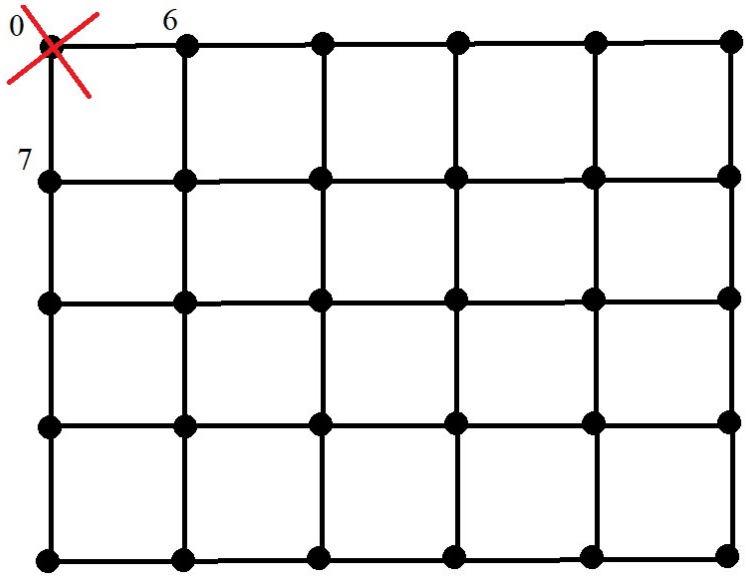
Варіант №9

Завдання № 1.

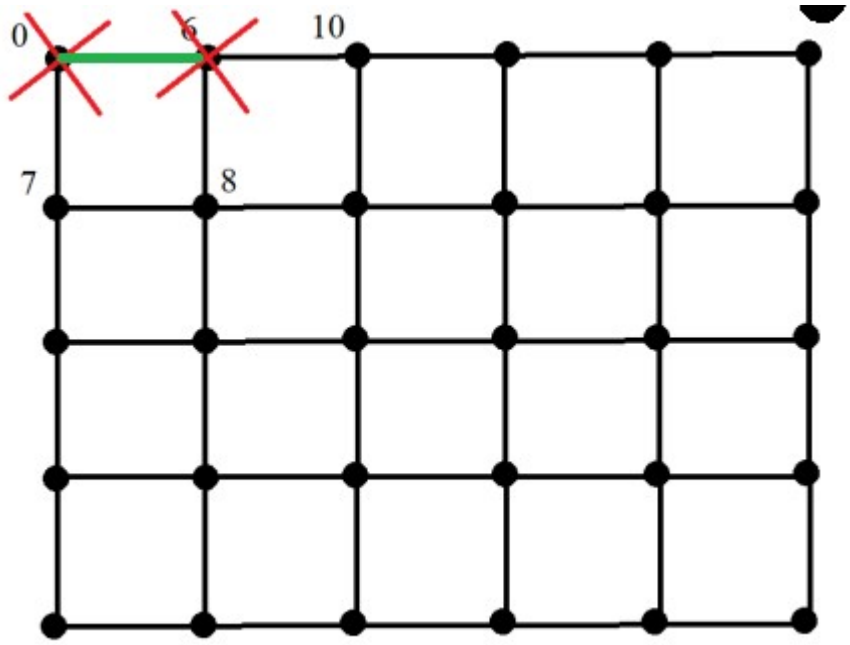
Розв'язати на графах наступні 2 задачі: 1. За допомогою алгоритму Дейкстра знайти найкоротший шлях у графі поміж парою вершин V0 і V * .



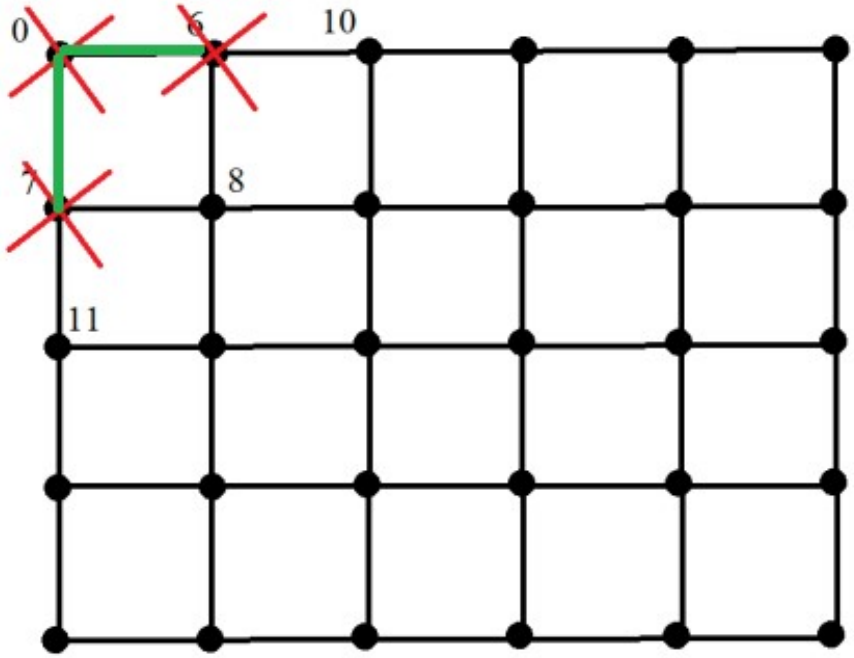
1.Починаємо з вершини V0. Знаходимо відстань до найближчих вершин та позначаємо вершину V0 опрацьованою. (Відстань до сусідньої вершини знаходимо як суму мінімальної відстані поточної і відстані від неї до цієї сусідньої , якщо ця відстань менша за мінімальну позначаємо її мінімальною , за умовчуванням мін.відстань рівна нескінченності)



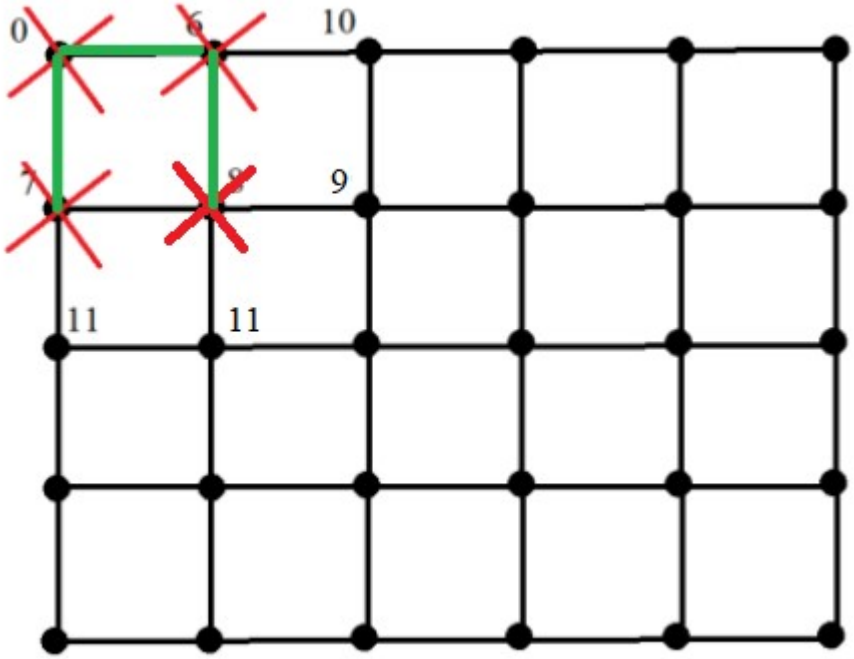
2. Переходимо до найближчої необробленої вершини. На даному кроці це вершина з відстанню до неї 6 , знаходимо відстані до її сусідніх вершин.

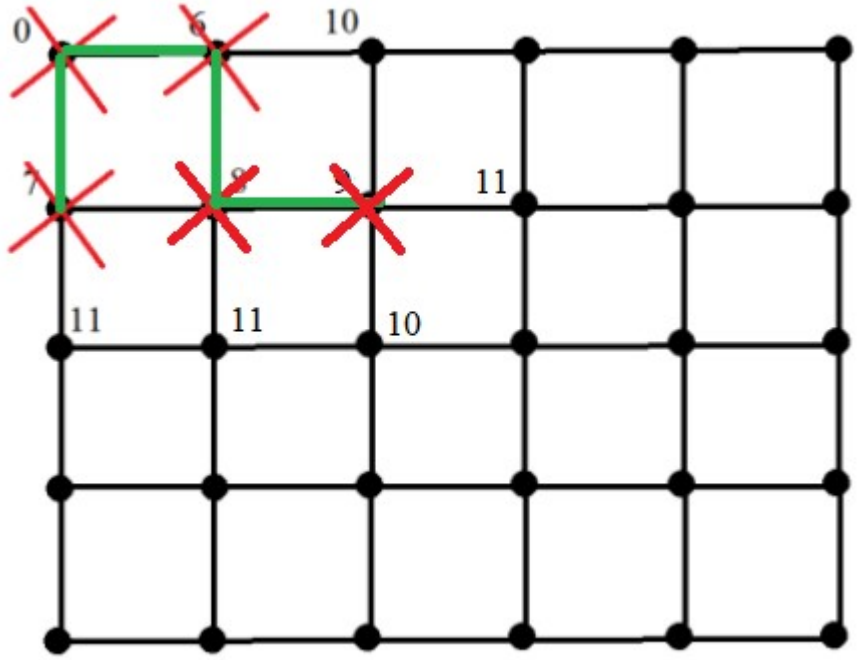


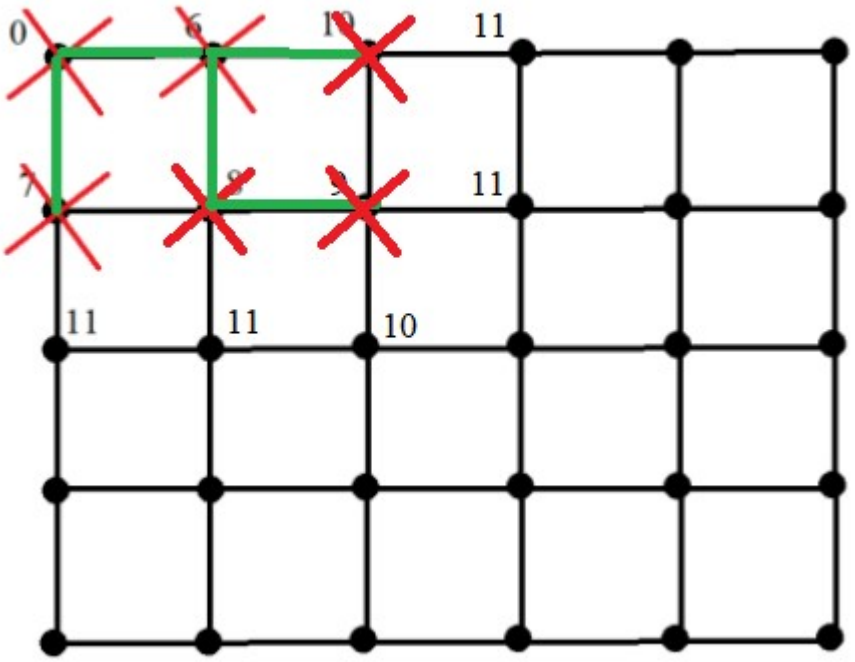
3. Наступною вершиною є вершина 7(Індексуватимемо вершини мінімальної відстаню до них, якщо існує декілька однакових вносимо додаткові індекси)

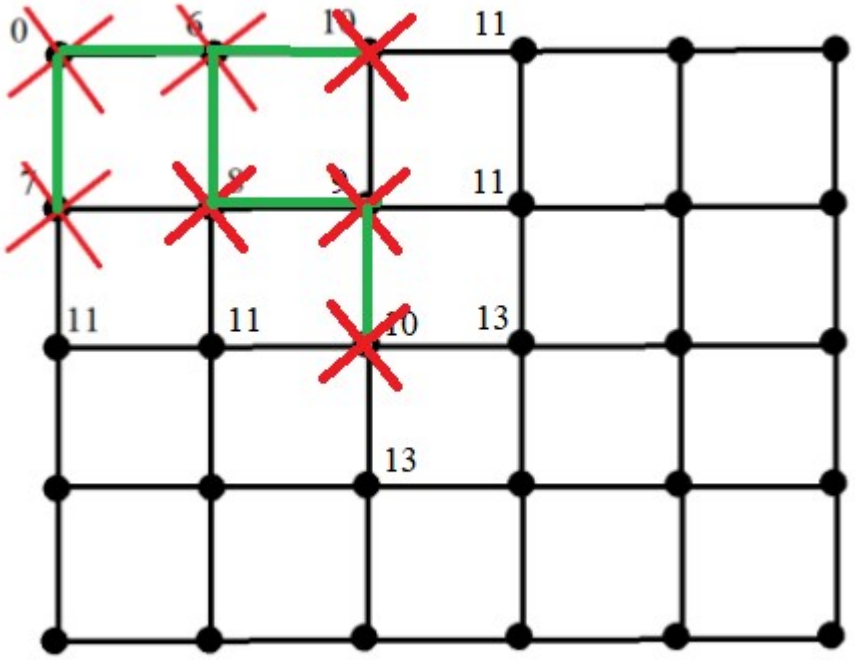


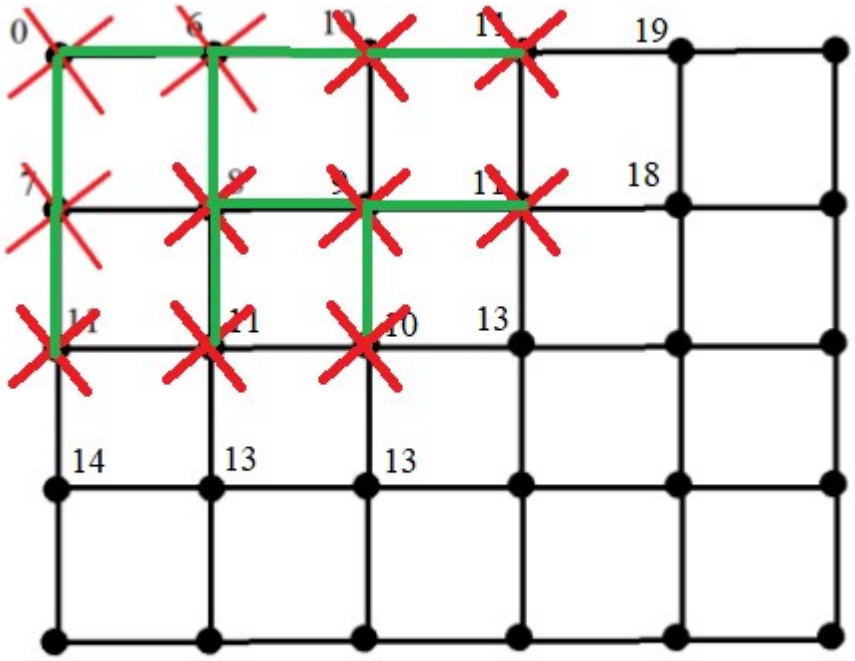
Аналогічно опрацьовуємо всі інші вершини

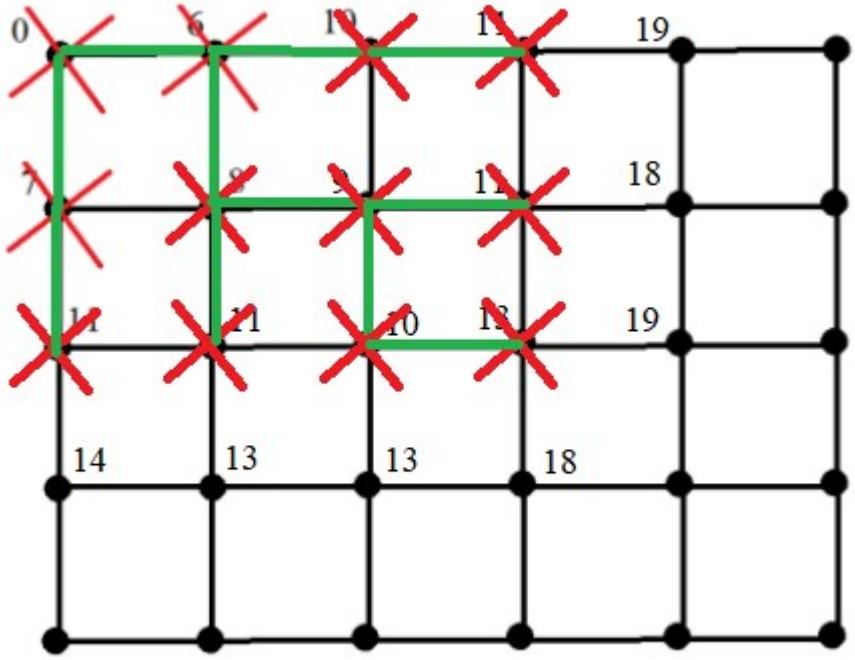


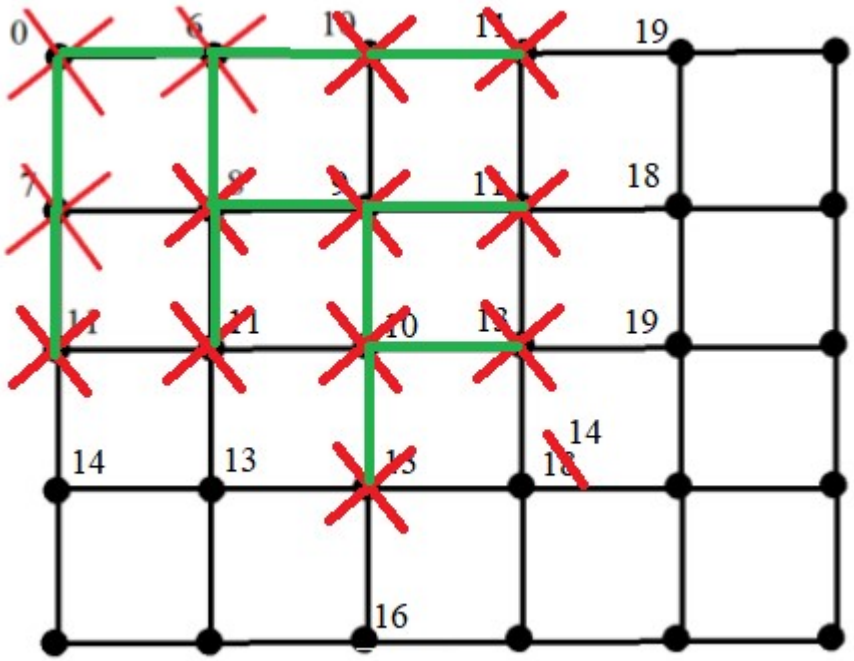


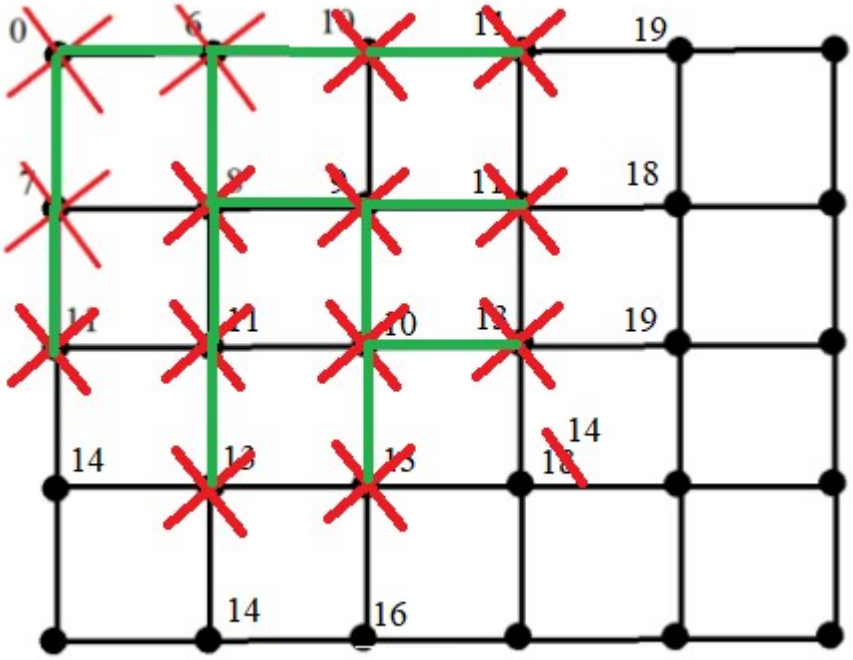

0
11
11
11
11
10

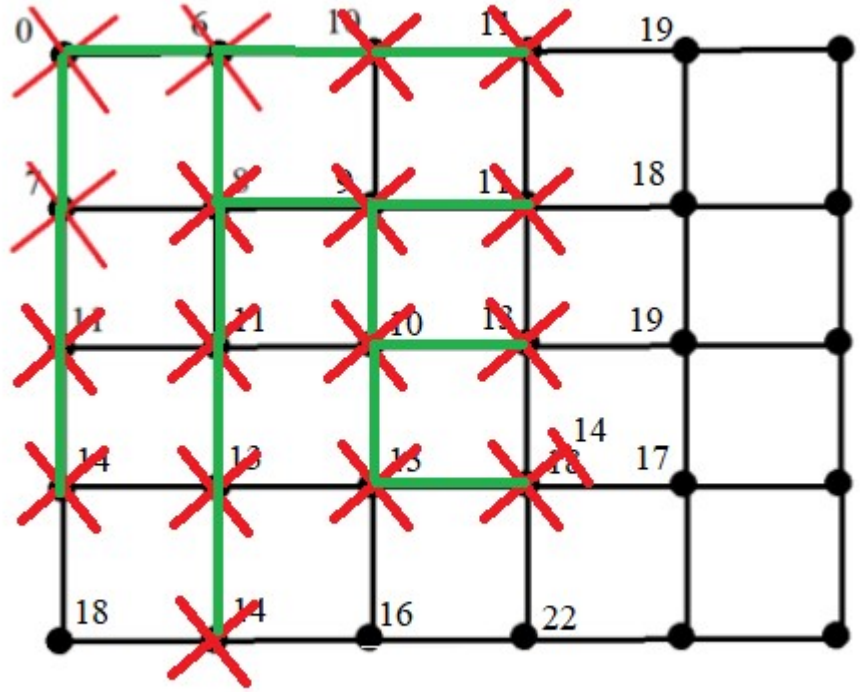

0
11
11
11
11
13
10
13

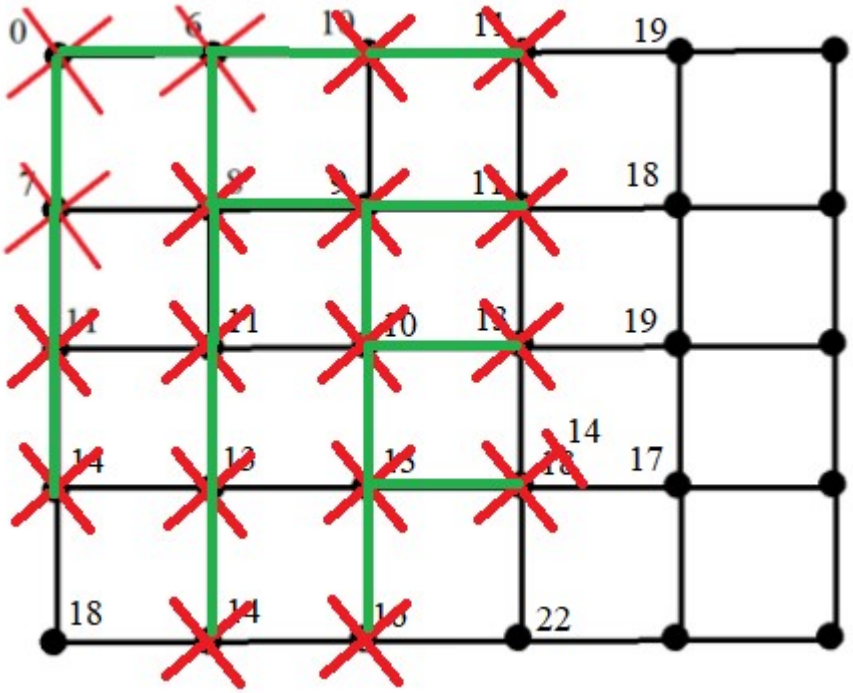

0
19
18
13
14
13
13

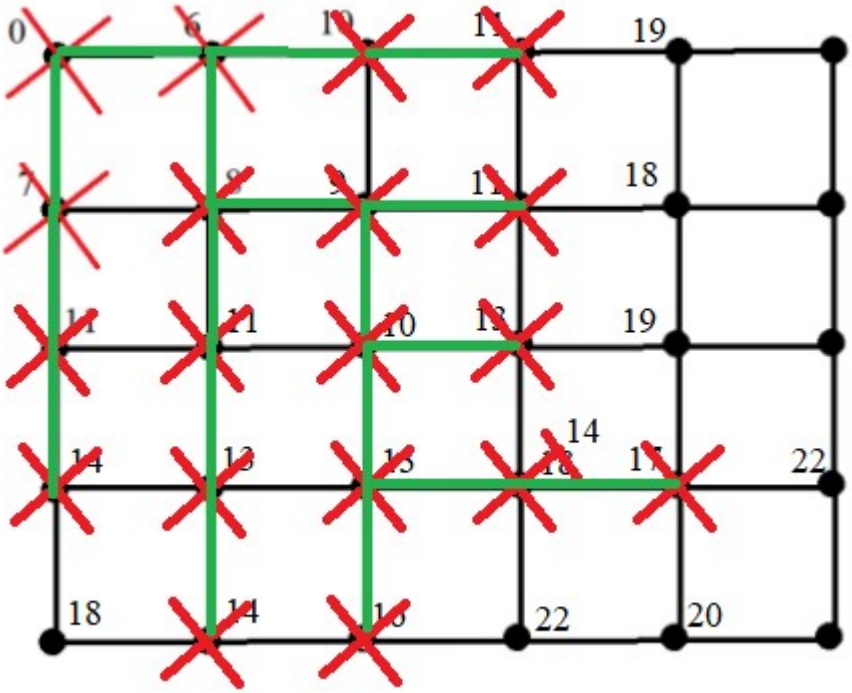

19
18
19
14
13
13
18

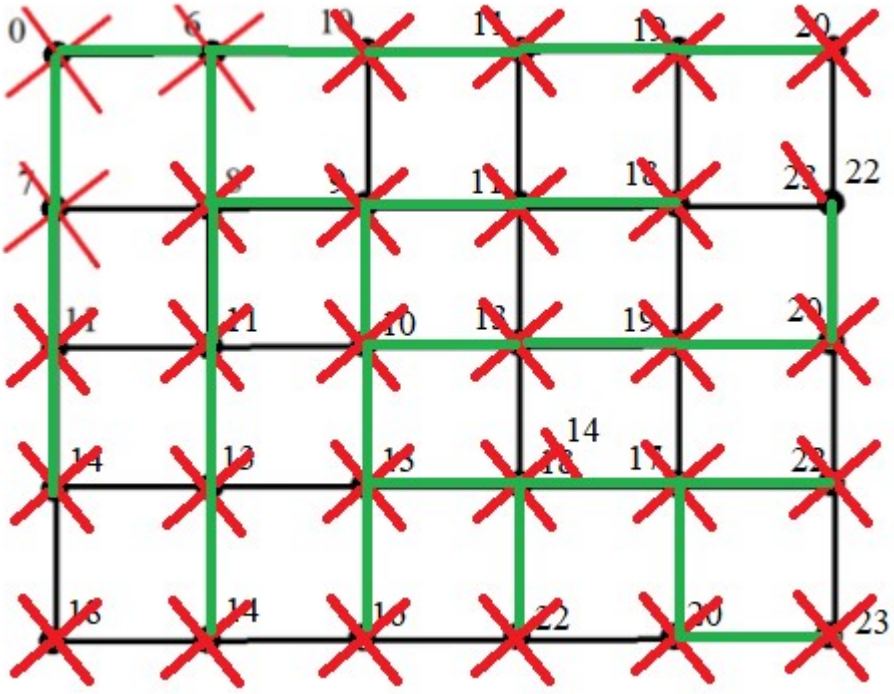

0
19
18
19
14
14
13
16


19
18
19
14
14
14
16

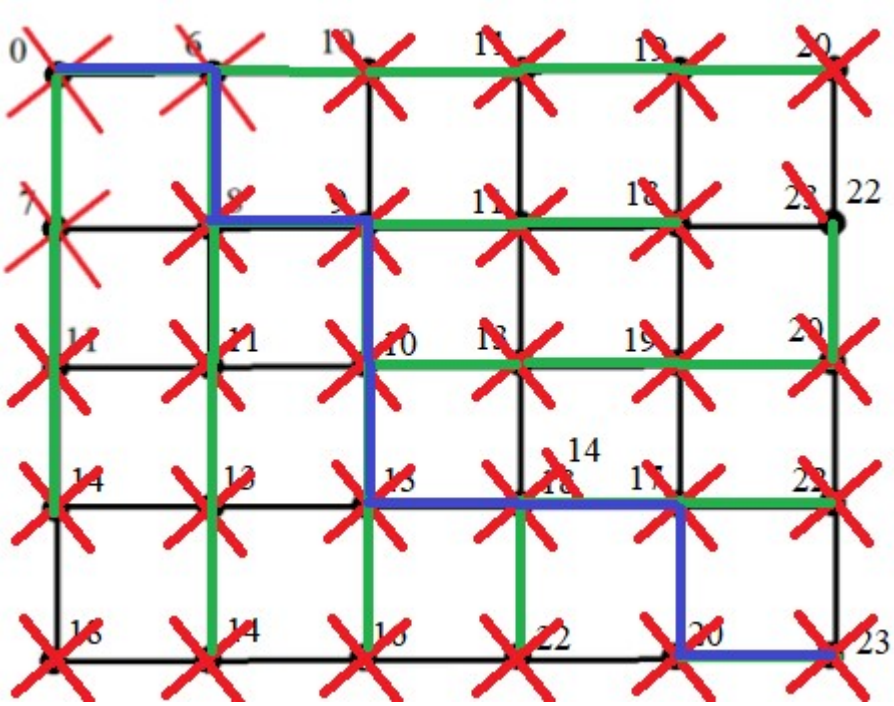

0
19
18
19
14
17
18
14
16
22


0
19
18
19
14
17
18
22


0
19
18
19
14
22
18
22
20

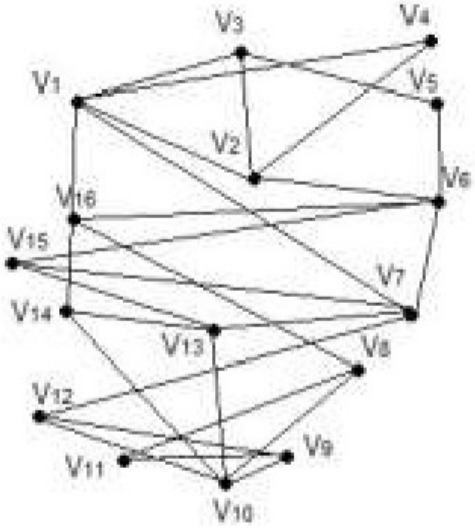

22
14
22
23

Найкоротший шлях :

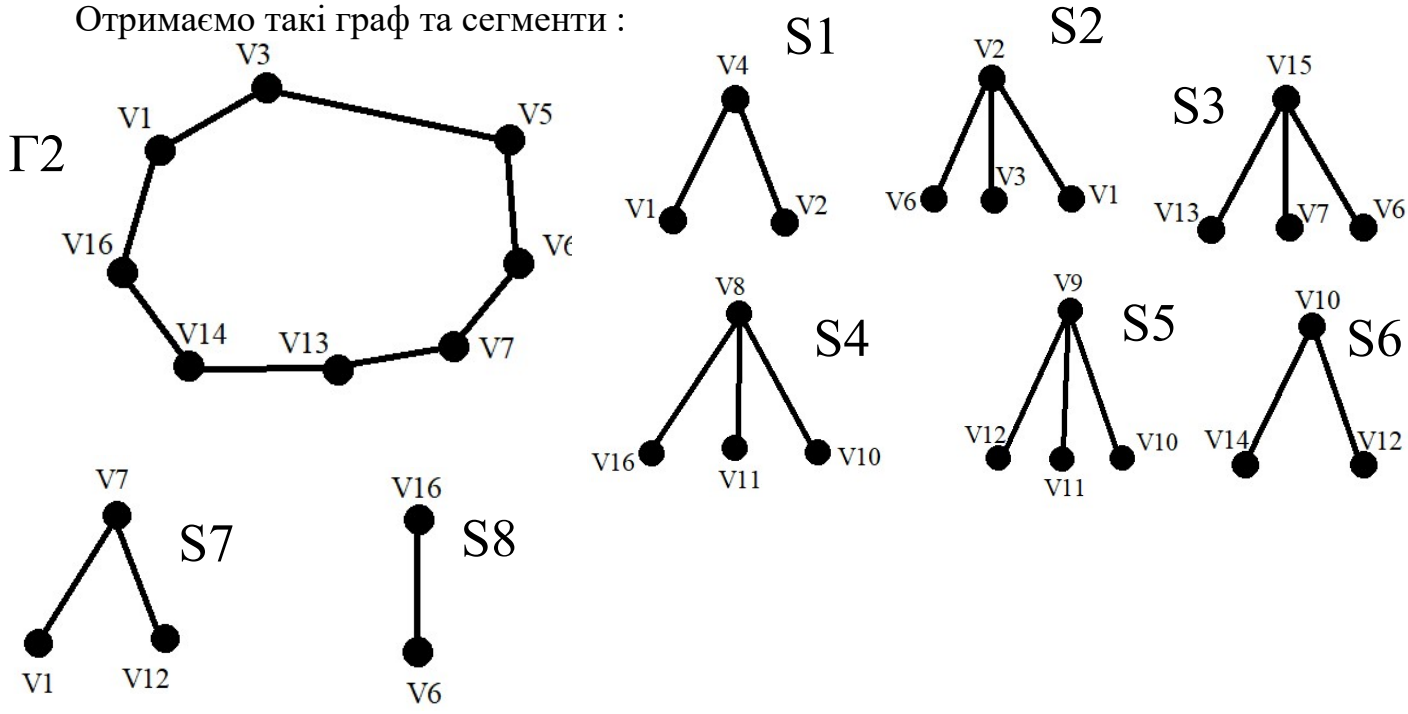


Мінімальна відстань - 23.

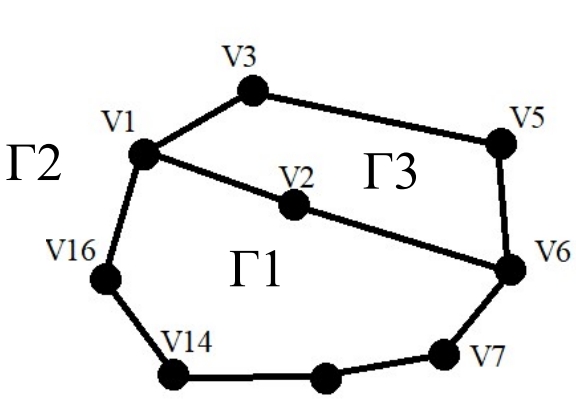
2. За допомогою гамма -алгоритма зробити укладку графа у площині, або довести що вона неможлива.

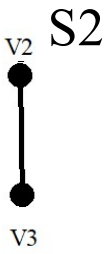


1.Укладемо спочатку цикл C=[V1,V3,V5,V6,V7,V13,V14,V16,V1], що розбиває площину на дві грані Г1 і Г2.
Отримаємо такі граф та сегменти :

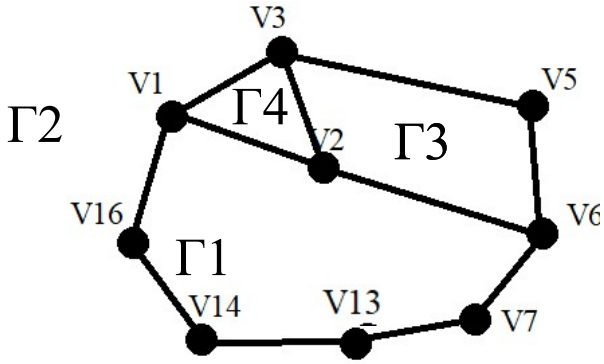


2.Помістимо сегмент альфа-ланцюг (V1,V2,V6) S2 в грань Г3



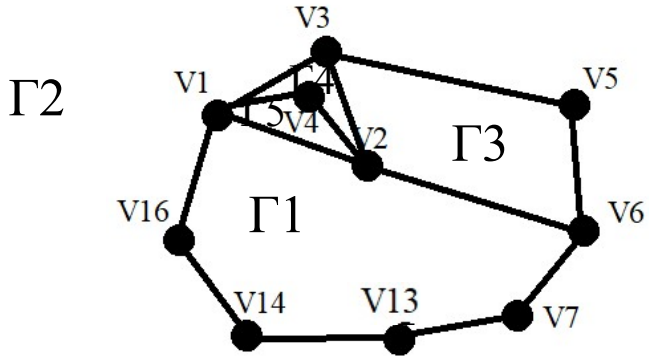


В утворену грань Г3 укладемо ланцюг (V2,V3)

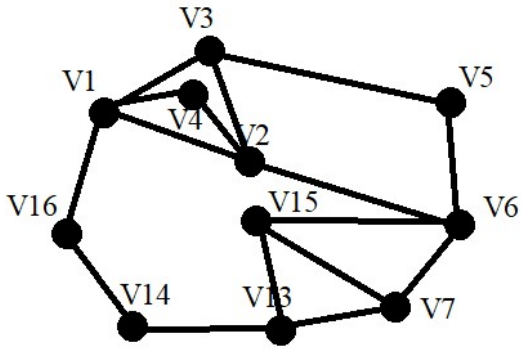


На даному етапі ми помістили весь сегмент S2 в граф G1

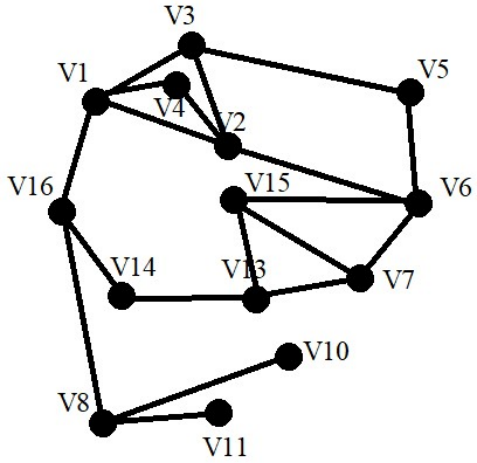
3. Тепер помістимо ланцюг (V1,V4,V2) сегмента S1



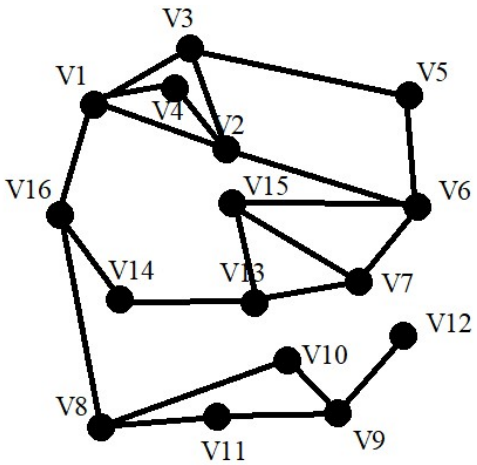
4. Виконуємо аналогічні кроки для всіх інших сегментів
Після додавання S3



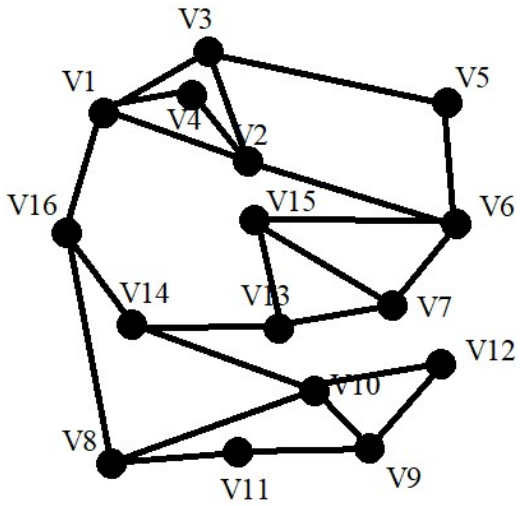
Після додавання S4



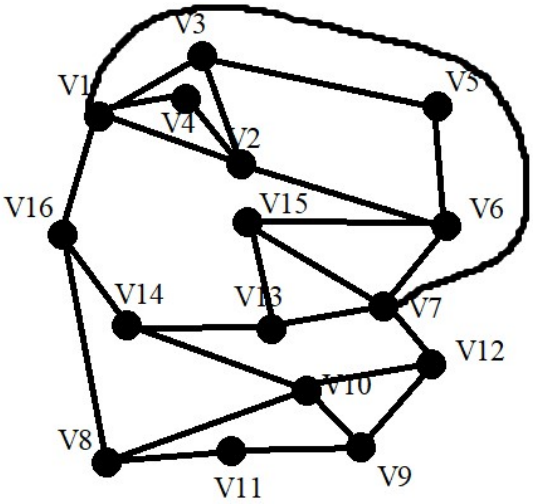
Після додавання S5



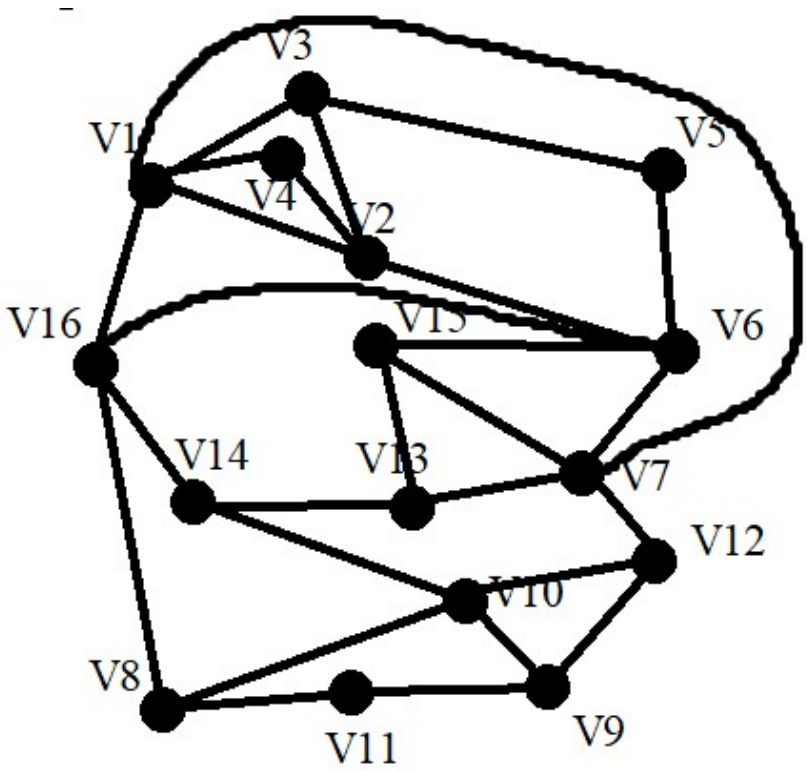
Після додавання S6



Після додавання S7

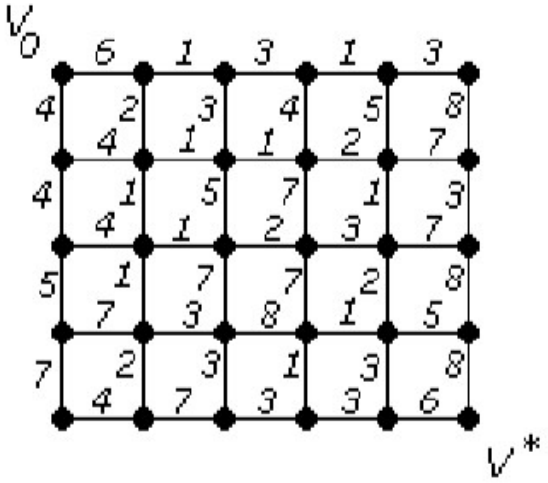


Після додавання S8



Отримали плоский граф ізоморфний до G.Отже граф G планарний.

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



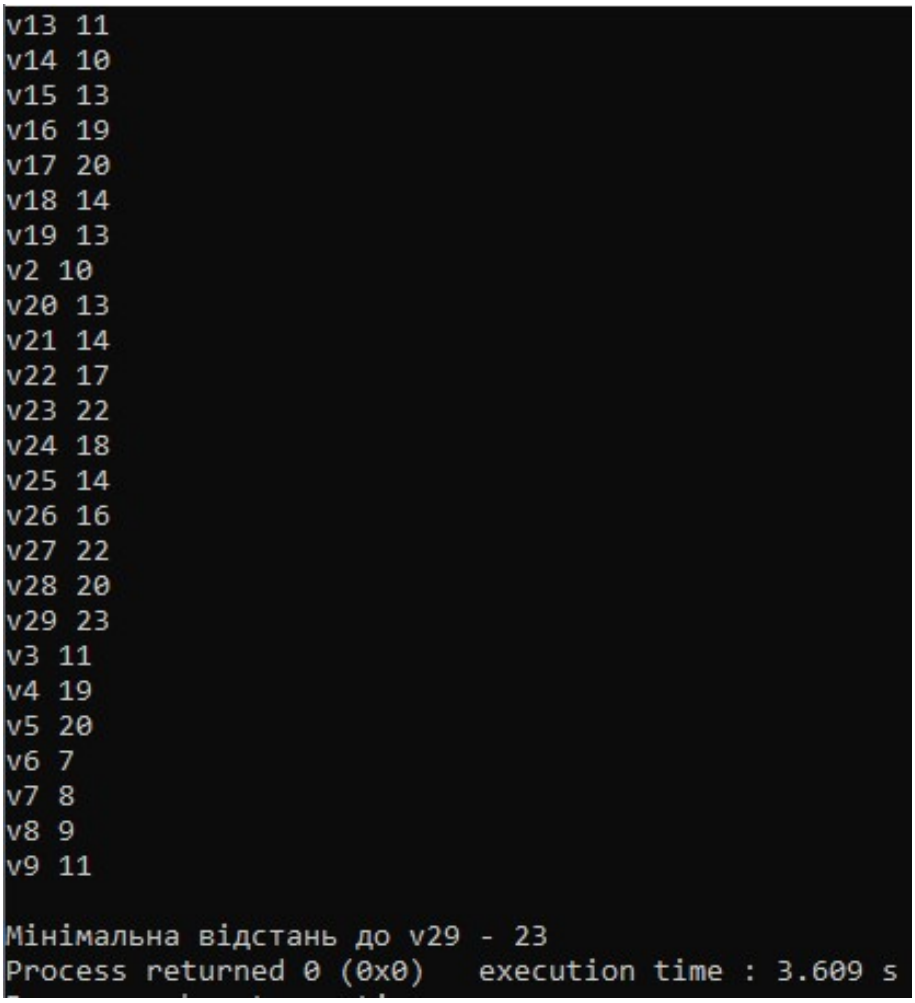
Код програми:

```
#include <iostream>
#include <windows.h>
#include <bits/stdc++.h>
using namespace std;
int consisting(string x,set <string> s){
int c = 0;
for(auto i : s){
	if (x == i)
		c = 1;
}
if (c ==1)
	return 1;
else
	return 0;
}
class edge {
	public :
		string vertices_1;
		string vertices_2;
		int weight;
};
int main()
{
	SetConsoleOutputCP(CP_UTF8);
	set < string > vertices;

	char *p;
```

```
	char * s = new char[255];

	map <string , int > vertice_weight;
	set <string> ban;
	gets(s);
	p = strtok(s," ");
	vertices.insert(p);
	string current = p;

	while(p){
		vertices.insert(p);
		vertice_weight.insert(pair <string,int>(p,-1));
		p = strtok(NULL," ");

	}
	vector <edge> edges;
	while(1){
		edge x;
		cin>>x.vertices_1;
		if (x.vertices_1=="end")
			break;
		cin>>x.vertices_2>>x.weight;
		edges.push_back(x);
	}

//	string current = "v0";
	vertice_weight["v0"] = 0;
	int x = 0;

	while(x < vertices.size()){
		for (auto edge : edges){
			if ((edge.vertices_1 == current)&&(!consisting(edge.vertices_2,ban))){
				for (auto p = vertice_weight.begin();p != vertice_weight.end();
p++){
					if (vertice_weight[edge.vertices_2] == -1){
						vertice_weight[edge.vertices_2] = edge.weight +
vertice_weight[current];
					}
					else if ( edge.weight+vertice_weight[current] <
vertice_weight[edge.vertices_2] )
						vertice_weight[edge.vertices_2] = edge.weight +
vertice_weight[current];

				}
			}
			if ((edge.vertices_2 == current)&&(!consisting(edge.vertices_1,ban))){
				for (auto p = vertice_weight.begin();p != vertice_weight.end();
```

```
p++){
                        if (vertice_weight[edge.vertices_1] == -1){
                                vertice_weight[edge.vertices_1] = edge.weight +
vertice_weight[current];
                        }
                        else if ( edge.weight +vertice_weight[current]<
vertice_weight[edge.vertices_1] )
                                vertice_weight[edge.vertices_1] =
edge.weight+vertice_weight[current];

                    }
                }


            }
            ban.insert(current);
            cout<<current<<" ";
            int mini = 9999999999;
            for(auto v : vertice_weight){
                if ((v.second < mini)&&(!consisting(v.first ,ban))&&(v.second != -1)){
                    mini = v.second;
                    current = v.first;
                }
            }
            x++;
        }
        cout<<endl;
        for (auto i : vertice_weight)
            cout<<i.first<<" "<<i.second<<endl;
        cout<<endl;
        cout<<"Мінімальна відстань до v29 - "<<vertice_weight["v29"];
        return 0;
}
```

Результат роботи програми:



Висновок: навчився знаходити мінімальний шлях і відстань до вершин алгоритмом Дейкстри , укладати граф на площині.