

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТУ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту

Лабораторна робота №5

з дисципліни
«Дискретна математика»

Виконав:

студент групи КН-113

Костів Богдан

Викладач: Мельникова Н.І.

Львів – 2019 р.

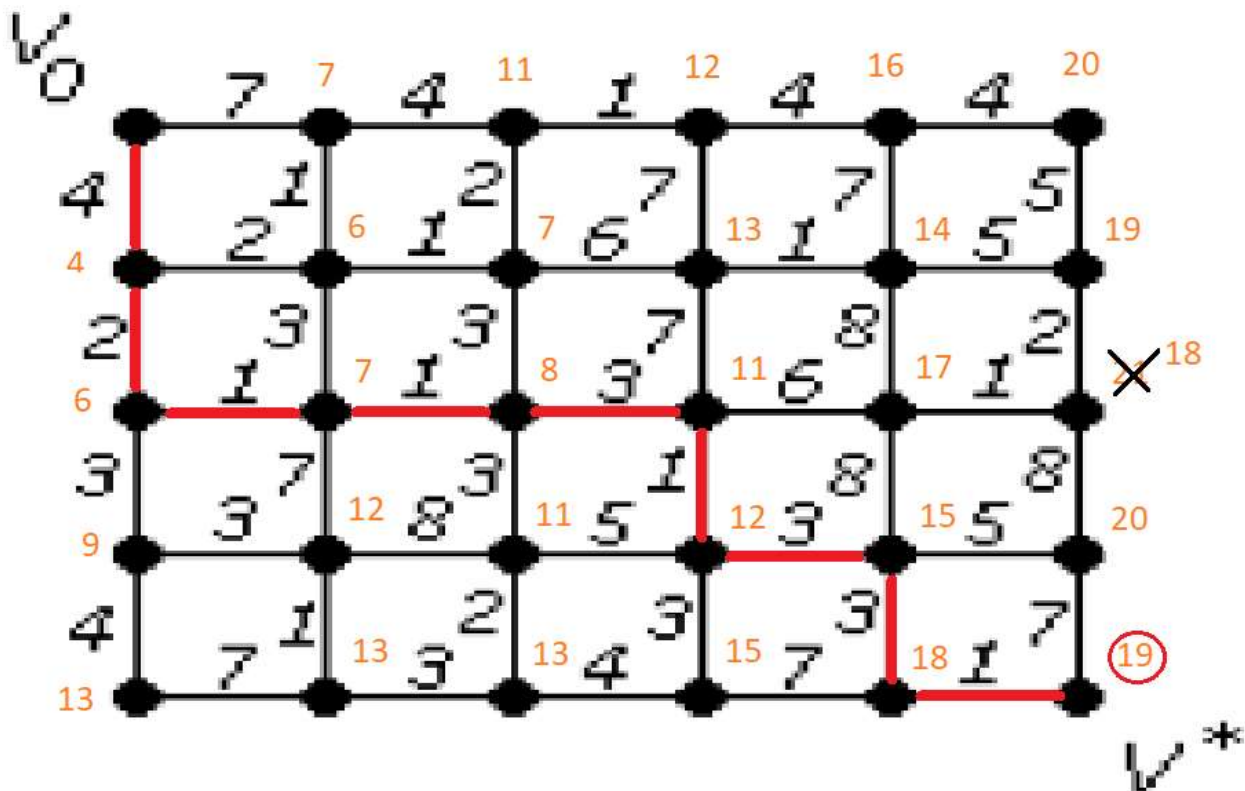
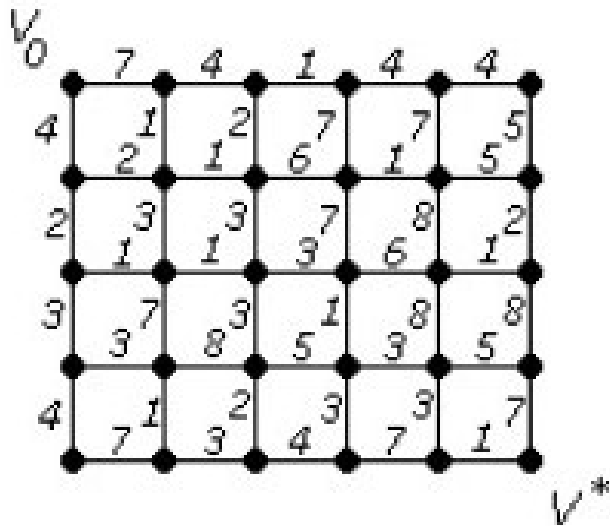
Лабораторна робота № 5.

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

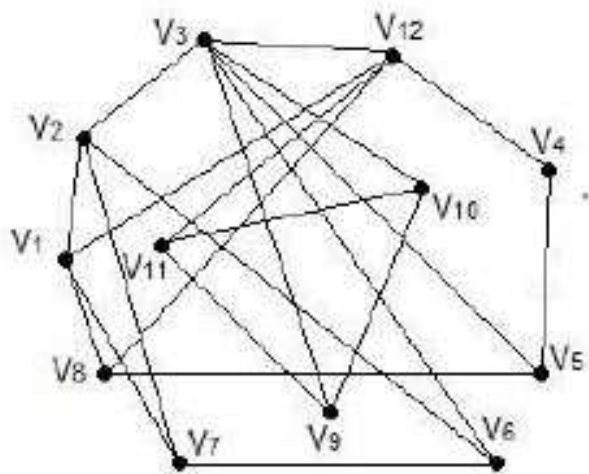
Мета роботи: набуття практичних вмінь та навичок з використання алгоритму Дейкстри.

Варіант №4

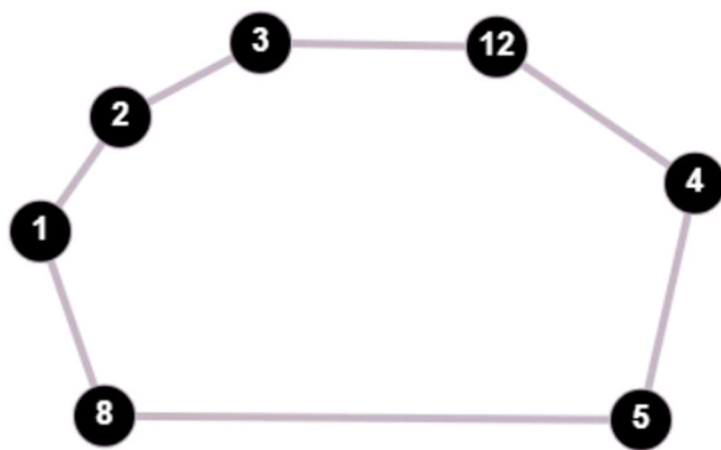
1. За допомогою алгоритму Дейкстри знайти найкоротший шлях у граfi поміж парою вершин V_0 і V^* .



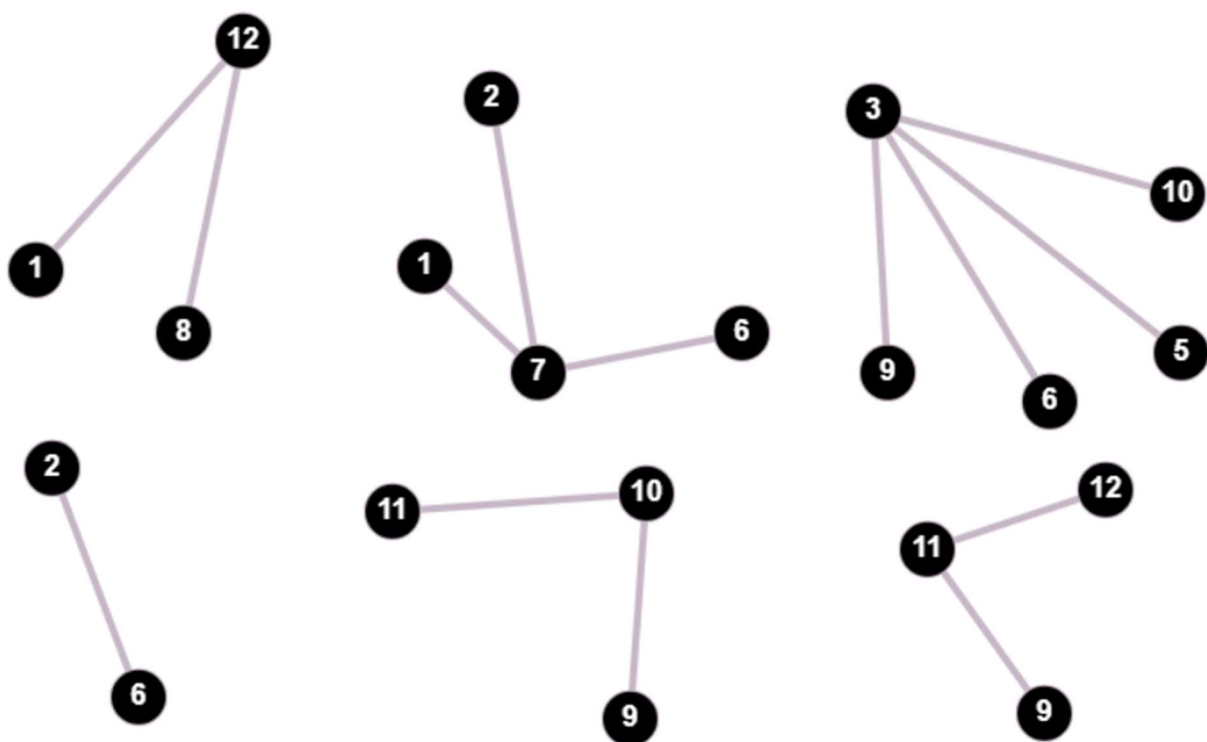
2. За допомогою γ -алгоритма зробити укладку графа у площині, або довести що вона неможлива.



Вибираємо довільний цикл з графа:

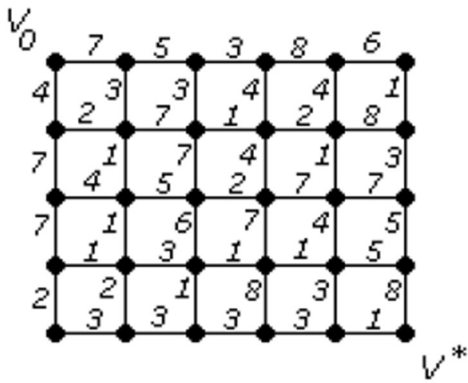


Сегменти:



A graph with 8 nodes labeled 1, 2, 3, 4, 5, 8, 12, and Γ_1 . The nodes are arranged in a roughly circular pattern. Edges connect nodes (1,2), (2,3), (3,12), (12,4), (4,5), (5,8), (8,1), and (8,12). There are also edges between (1,12) and (8,4). The labels Γ_1 , Γ_2 , Γ_3 , and Γ_4 are placed near the edges (12,4), (4,5), (1,2), and (2,3) respectively.

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



```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <fstream>
#include <locale>

using namespace std;

int main()
{
    ifstream fin("MyFile.txt");
    setlocale(LC_ALL, "Ukrainian");
    int vershina, rebra;
    fin >> vershina >> rebra;
    const int SIZE = 30;
    int matrix[SIZE][SIZE]; // матриця зв'язків
    int distance[SIZE]; // мінімальна відстань
    int visited[SIZE]; // чи відвідані вершини
    int dis, top, min;
    int begin_index = 0;

    for (int i = 0; i < vershina; i++) // Ініціалізація матриці зв'язків
    {
        distance[i] = 99999;
        visited[i] = 0;
        for (int j = 0; j < vershina; j++)
        {
            matrix[i][j] = 0;
            matrix[j][i] = 0;
        }
    }
    for (int i = 0; i < rebra; i++) {
        int v1, v2, dis;
        fin >> v1 >> v2 >> dis;
        matrix[v1 - 1][v2 - 1] = dis;
        matrix[v2 - 1][v1 - 1] = dis;
    }
    cout << "DIJKSTRA
ALGORITHM\n~~~~~\n\n";

    distance[0] = 0;
    do {
        top = 99999;
        min = 99999;
        for (int i = 0; i < vershina; i++)
        {
            if (visited[i] == 0 && distance[i] < min)
            {
                min = distance[i];
            }
        }
    } while (min < 99999);
}
```

```

top = i;
}

}
if (top != 99999)
{
for (int i = 0; i < vershina; i++)
{
if (matrix[top][i] > 0)
{
dis = min + matrix[top][i];
if (dis < distance[i])
{
distance[i] = dis;
}
}
}
visited[top] = 1;
}
} while (top < 99999);

int end = vershina - 1;
int waga = distance[end];
int way[30];
way[0] = vershina - 1;
int k = 1;
while (end != 0)
{
for (int i = 0; i < vershina; i++)
{
if (matrix[end][i] > 0)
{
if (distance[i] == waga - matrix[end][i])
{
waga = distance[i];
way[k] = i;
end = i;
k++;
}
}
}
}
cout << "\nSmallest path from V0 to V29: ";
for (int i = k; i > 0; i--)
{
if (i - 1 > 0)
cout << way[i - 1] << " -> ";
else
cout << way[i - 1];
}
cout << "\n\nDistance length: " << distance[vershina - 1] << endl;
cout <<
"\n\n~~~~~\n\n";
}

```

DIJKSTRA ALGORITHM

~~~~~

Smallest path from V0 to V29: 0 -> 6 -> 7 -> 13 -> 19 -> 20 -> 21 -> 22 -> 28 -> 29

Distance length: 17

~~~~~

Висновок: в результаті проведеної роботи ми ознайомились із знаходженням найкоротшого маршруту за алгоритмом Дейкстри та побудовою плоских планарних графів.