

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ  
УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»  
Кафедра штучного інтелекту

Звіт  
із лабораторної роботи №5  
із дисципліни “Дискретна математика”  
Варіант №15

Виконав:

Студент групи КН-113

Черній Ю.М.

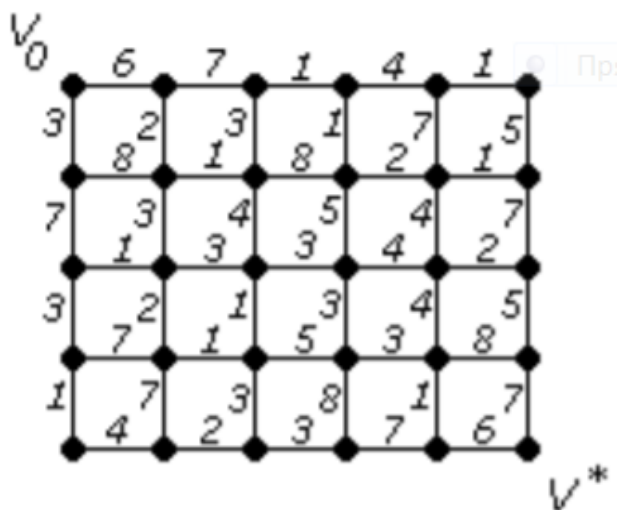
Викладач:

Мельникова Н.І

# Завдання № 1.

Розв'язати на графах наступні 2 задачі:

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



Розв'язання.

Відстань до всіх вершин графа рівне нескінченності. Відстань до  $V_0 = 0$ . Жодна вершина графа ще не опрацьована.

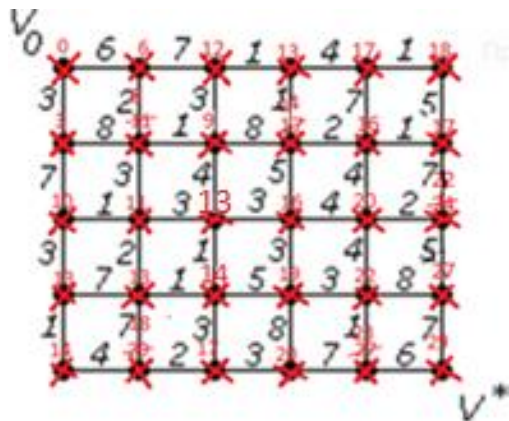
Знаходимо таку вершину (із ще не оброблених), поточна найкоротша відстань до якої мінімальна. В нашому випадку це вершина  $V_0$ . Обходимо всіх її сусідів і, якщо шлях в сусідню вершину через  $V_0$  менший за поточний мінімальний шлях в цю сусідню вершину, то запам'ятовуємо цей новий, коротший шлях як поточний найкоротший шлях до сусіда.

Отже запишемо, що шлях від  $V_0$  до  $V_{01}$  рівне 6, а від  $V_0$  до  $V_{10}$  рівне 3. Виберемо мінімальний. Ним є 3 – шлях від  $V_0$  до  $V_{10}$ .

Всі сусіди вершини  $V_0$  перевірені. Поточна мінімальна відстань до вершини  $V_0$  вважається остаточною і обговоренню не підлягає. Тому точку  $V_0$  може викреслити.

Знову знаходимо найближчу невикреслену вершину з точок  $V_{10}$  та  $V_0$ , зменшуючи відстань до усіх сусідніх вершин.

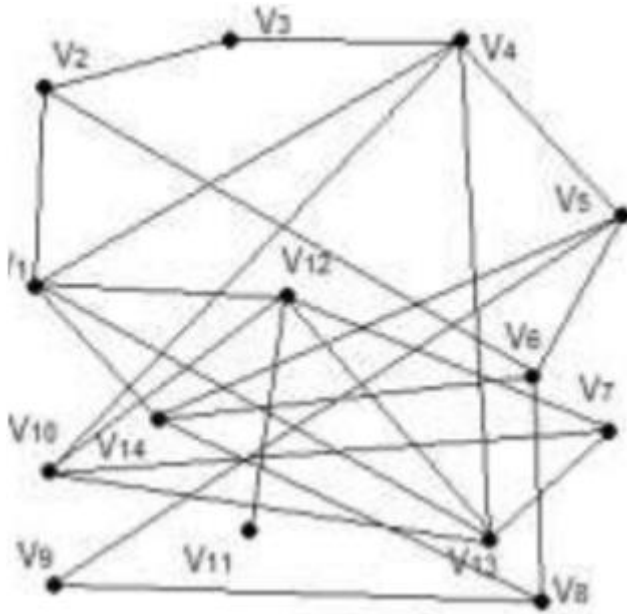
Продовжуємо виконувати ці дії допоки не викреслені всі вершини.



Отже найкоротший шлях у графі поміж парою вершин  $V_0$  і  $V^*$  рівний 29.

1->2->8->9->15->16->22->23->29->30.

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



Розв'язання.

Граф укладається лише тоді, коли він планарний.

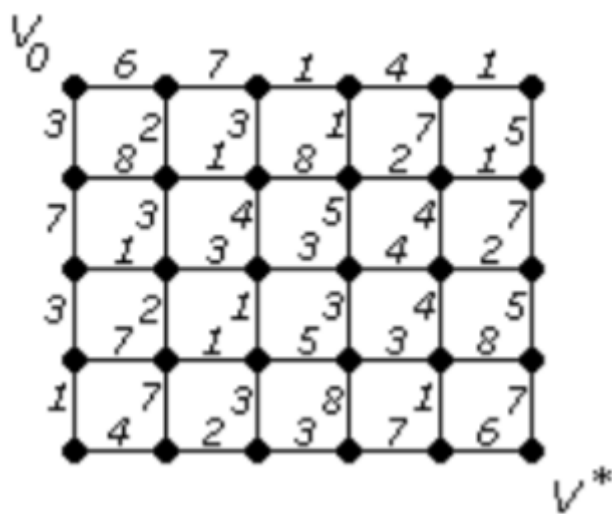
Критерій непланарності:

- достатня умова — якщо граф містить дводольний підграф  $K_{3,3}$  або повний підграф  $K_5$ , то він є не планарним;
- необхідна умова — якщо граф не планарний, то він повинен містити більше 4 вершин, степінь яких більше 3, або більше 5 вершин степеня 2.

Отже підрахуємо кількість вершин, степінь яких більше 3. Вершини  $V_4, V_1, V_{10}, V_{13}, V_5$  мають більше 3.

Отже граф не укладається.

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



Текст програми:

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
    setlocale(LC_CTYPE, "ukr");
    bool visited[30]; int tmp, l;
    for (int i = 0; i < 30; i++)
    {
        visited[i] = false;
    }
    int mas1[30][30];
    int min[30], min1[30], min2[30];
    for (int i = 0; i < 30; i++)
    {
        min[i] = 999;
    }
    ifstream in("Input.txt");
    for (int i = 0; i < 30; i++)
    {
        for (int k = 0; k < 30; k++)
        {
            in >> mas1[i][k];
            if (mas1[i][k] == 0)
            {
                mas1[i][k] = 999;
            }
        }
    }
    int e, e1;
    cout << "Введіть початкову точку: ";
    cin >> e;
    l = e-1;
    cout << "Введіть кінцеву точку: ";
    cin >> e1;
    min[l] = 0;
    mas1[l][l] = 0;
    for (int u = 0; u < 30; u++)
    {
        for (int i = 0; i < 30; i++)
        {
            if ((min[i] > mas1[l][i]) && (visited[i] == false))
            {
                if (min[i] > (mas1[l][i] + min[l]))
                {
                    min[i] = mas1[l][i] + min[l];
                }
            }
        }
        visited[l] = true;
        for (int i = 0; i < 30; i++)
        {
            if (visited[i] == true)
            {
                min2[i] = 999;
            }
            else { min2[i] = min[i]; }
        }
        for (int j = 0; j < 29; j++) {
            for (int i = 0; i < 29; i++) {
                if (min2[i] > min2[i + 1])
                {
                    tmp = min2[i];
                    min2[i] = min2[i + 1];
                    min2[i + 1] = tmp;
                }
            }
        }
    }
}
```

}

[illegible]

000070000201000040000000000000  
000005000010000007000000000000  
000000700000010000300000000000  
000000030000103000020000000000  
000000004000030300001000000000  
000000000500003040000300000000  
000000000040000402000040000000  
000000000007000020000005000000  
000000000000300000070000100000  
000000000000020000701000070000  
000000000000001000010500003000  
000000000000000300005030000800  
000000000000000040000308000010  
000000000000000005000080000007  
000000000000000000100000040000  
000000000000000000070000402000  
000000000000000000003000020300  
000000000000000000000800003070  
000000000000000000000010000706  
000000000000000000000007000060

## Висновок

Набув практичних вмінь та навичок з використання алгоритму Дейкстри.